Arbitrary-Lagrangian-Eulerian One-Step WENO Finite Volume Schemes on Unstructured Triangular Meshes

Walter Boscheri[1], Michael Dumbser[1,*]

[1] *Laboratory of Applied Mathematics, Department of Civil, Environmental and Mechanical Engineering, University of Trento, I-38123 Trento, Italy*

Abstract. In this article we present a new class of high order accurate Arbitrary–Eulerian–Lagrangian (ALE) one–step WENO finite volume schemes for solving nonlinear hyperbolic systems of conservation laws on moving two dimensional unstructured triangular meshes. A WENO reconstruction algorithm is used to achieve high order accuracy in space and a high order one–step time discretization is achieved by using the local space–time Galerkin predictor proposed in [25]. For that purpose, a new element–local weak formulation of the governing PDE is adopted on moving space–time elements. The space-time basis and test functions are obtained considering Lagrange interpolation polynomials passing through a predefined set of nodes. Moreover, a polynomial mapping defined by the same local space–time basis functions as the weak solution of the PDE is used to map the moving physical space–time element onto a space–time reference element. To maintain algorithmic simplicity, the final ALE one–step finite volume scheme uses moving triangular meshes with *straight* edges. This is possible in the ALE framework, which allows a local mesh velocity that is different from the local fluid velocity. We present numerical convergence rates for the schemes presented in this paper up to sixth order of accuracy in space and time and show some classical numerical test problems for the two–dimensional Euler equations of compressible gas dynamics.

Key words: Arbitrary Lagrangian-Eulerian, high order reconstruction, WENO, finite volume, local space-time Galerkin predictor, moving unstructured meshes, Euler equations

1 Introduction

In this paper we present a new family of high order accurate Lagrangian–type one–step finite volume schemes for solving nonlinear hyperbolic balance laws, with non stiff algebraic source term. The main advantage of working in a Lagrangian framework is that material interfaces can be identified and located precisely, since the computational mesh

---

*Corresponding author. *Email addresses:* `walter.boscheri@unitn.it` (W. Boscheri),
`michael.dumbser@unitn.it` (M. Dumbser)

is moving with the local fluid velocity, hence obtaining a more accurate resolution of material interfaces. For this reason a lot of research has been carried out in the last decades in order to develop Lagrangian methods, whose algorithms can start either directly from the conservative quantities such as mass, momentum and total energy [57,68], or from the nonconservative form of the governing equations, as proposed in [6,9,79]. Furthermore we can split the existing Lagrangian schemes into two main classes, depending on the location of the physical variables on the mesh: in one case the velocity is defined at the cell interfaces and the other variables at the cell barycenter, hence adopting a *staggered mesh* approach, while in the other case all variables are defined at the cell barycenter, therefore using a *cell–centered* approach.

The equations of Lagrangian gas dynamics have been considered in [60], where several different Godunov-type finite volume schemes have been presented and where a new Roe linearization has been introduced in order to define proper estimates of the maximum signal speeds in HLL–type Riemann solvers in Lagrangian coordinates. A cell-centered Godunov scheme has been proposed by Carré et al. [10] for Lagrangian gas dynamics on general multi-dimensional unstructured meshes. In this case the finite volume scheme is node based and compatible with the mesh displacement. In [21] Després and Mazeran introduce a new formulation of the multidimensional Euler equations in Lagrangian coordinates as a system of conservation laws associated with constraints. Furthermore they propose a way to evolve in a coupled manner both the physical and the geometrical part of the system [22], writing the two–dimensional equations of gas dynamics in Lagrangian coordinates together with the evolution of the geometry as a weakly hyperbolic system of conservation laws. This allows the authors to design a finite volume scheme for the discretization of Lagrangian gas dynamics on moving meshes, based on the symmetrization of the formulation of the physical part. In a recent work Després et al. [18] propose a new method designed for cell-centered Lagrangian schemes, which is translation invariant and suitable for curved meshes. General polygonal grids are also considered by Maire et al. [54–56], who develop a general formalism to derive first and second order cell-centered Lagrangian schemes in multiple space dimensions. By the use of a node-centered solver [56], the authors obtain the time derivatives of the fluxes. The solver may be considered as a multi-dimensional extension of the Generalized Riemann problem methodology introduced by Ben-Artzi and Falcovitz [5], Le Floch et al. [8,41] and Titarev and Toro [70,71,73]. So far, all the above–mentioned schemes are at most second order accurate in space and time.

In order to achieve higher accuracy, Cheng and Shu were the first who introduced a high order essentially non-oscillatory (ENO) reconstruction in Lagrangian schemes [14,53]. They developed a class of cell centered Lagrangian finite volume schemes for solving the Euler equations, using both Runge-Kutta and Lax-Wendroff-type time stepping to achieve also higher order in time. Furthermore a formalism for symmetry preserving Lagrangian schemes has been proposed by Cheng and Shu, see [15,16]. The higher order schemes presented in [14,53] were the first better than second order non–oscillatory Lagrangian–type finite volume schemes ever proposed. Higher order finite

element methods on unstructured meshes have been investigated in [62, 67]. In a very recent paper Dumbser et al. [30] propose a new class of high order accurate Lagrangian–type one-step WENO finite volume schemes for the solution of stiff hyperbolic balance laws. They consider the one–dimensional case and develop a Lagrangian algorithm up to eighth order of accuracy, based on high order WENO reconstruction in space and the local space–time discontinuous Galerkin predictor method proposed in [26] to obtain a high order one–step scheme in time.

Other schemes can be also included and mentioned as Lagrangian algorithms, e.g. meshless particle schemes that adopt a fully Lagrangian approach, such as the smooth particle hydrodynamics (SPH) method [36–39, 58], which can be used to simulate fluid motion in complex deforming domains. Also within the SPH approach, which is a fully Lagrangian method, the mesh moves with the local fluid velocity, whereas in Arbitrary Lagrangian Eulerian (ALE) schemes, see e.g. [13, 23, 34, 35, 44, 64, 68], the mesh moves with an arbitrary mesh velocity that does not necessarily coincide with the real fluid velocity. Furthermore one can find Semi-Lagrangian schemes, which are mainly used for solving transport equations [42, 66]. Here, the numerical solution at the new time level is computed from the known solution at the present time by following backward in time the Lagrangian trajectories of the fluid to the end-point of the trajectory. Since the end-point does not coincide with a grid point, an interpolation formula is required in order to evaluate the unknown solution, see e.g. [7, 11, 12, 20, 46, 52, 65]. For the sake of clarity we specify that in Semi-Lagrangian algorithms the mesh is fixed as in a classical Eulerian approach. An alternative to Lagrangian methods for the accurate resolution of material interfaces has been developed in the Eulerian framework on fixed meshes under the form of the ghost–fluid method [32, 33, 40], together with a level–set approach [59, 63], where the level–set function represents the signed distance from the material interface and its zeros locate the interface position.

In this paper we introduce a new better than second order accurate *two–dimensional* Lagrangian–type one–step WENO finite volume scheme on *unstructured triangular meshes*: high order of accuracy in space is obtained using a WENO reconstruction [2, 3, 27, 28, 45, 47, 72, 76, 80], although other higher order spatial reconstruction schemes could be adopted as well, see [1, 17]. High order accuracy in time is achieved with a local space–time Galerkin predictor, as introduced in [26, 30, 31, 43]. The method proposed in this article is presented as an Arbitrary–Lagrangian–Eulerian scheme in order to allow arbitrary grid motion. This allows us to use curved space–time elements in the local predictor stage but triangles delimited by straight line segments in the resulting one–step finite volume scheme (corrector stage). This choice has been made to maintain algorithmic simplicity.

The outline of this article is as follows: in Section 2 we present the details of the proposed numerical scheme, while in Section 3 we show numerical convergence rates up to sixth order of accuracy in space and time for a smooth problem as well as numerical results for several different test cases governed by the compressible Euler equations. The paper closes with some concluding remarks and an outlook to possible future extensions of the method in Section 4.

2 Numerical Method

In this article we consider general nonlinear systems of hyperbolic balance laws of the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) = \mathbf{S}(\mathbf{Q}), \qquad (x,y) \in \Omega(t) \subset \mathbb{R}^2, \quad t \in \mathbb{R}_0^+, \quad \mathbf{Q} \in \Omega_{\mathbf{Q}} \subset \mathbb{R}^\nu, \qquad (2.1)$$

where $\mathbf{Q} = (q_1, q_2, ..., q_\nu)$ is the vector of conserved variables defined in the space of the admissible states $\Omega_{\mathbf{Q}} \subset \mathbb{R}^\nu$, $\mathbf{F}(\mathbf{Q}) = (\mathbf{f}(\mathbf{Q}), \mathbf{g}(\mathbf{Q}))$ is the nonlinear flux tensor and $\mathbf{S}(\mathbf{Q})$ represents a nonlinear but non-stiff algebraic source term. The spatial position vector is denoted by $\mathbf{x} = (x,y)$, t is the time and $\Omega(t)$ is the time–dependent computational domain.

The two-dimensional time–dependent computational domain $\Omega(t)$ is discretized at the current time t^n by a set of triangular elements T_i^n. The union of all elements is called the *current triangulation* $\mathcal{T}_\Omega^n$ of the domain $\Omega(t^n) = \Omega^n$ and can be expressed as

$$\mathcal{T}_\Omega^n = \bigcup_{i=1}^{N_E} T_i^n, \qquad (2.2)$$

where N_E is the total number of elements contained in the domain.

In a Lagrangian framework we are dealing with a *moving* mesh, hence the elements deform while the solution is evolving in time. It is therefore convenient to adopt a *local* reference coordinate system $\xi - \eta$, where the reference element T_e is defined. The spatial mapping of the triangular elements T_i^n at the current time t^n from reference coordinates $\xi - \eta$ to physical coordinates $x - y$ is given by the relation

$$\begin{aligned} x &= X_{1,i}^n + \left(X_{2,i}^n - X_{1,i}^n \right) \xi + \left(X_{3,i}^n - X_{1,i}^n \right) \eta, \\ y &= Y_{1,i}^n + \left(Y_{2,i}^n - Y_{1,i}^n \right) \xi + \left(Y_{3,i}^n - Y_{1,i}^n \right) \eta, \end{aligned} \qquad (2.3)$$

where $\mathbf{X}_{k,i}^n = \left(X_{k,i}^n, Y_{k,i}^n \right)$ denotes the vector of spatial coordinates of the k-th vertex of the triangle T_i^n in physical coordinates at the current time t^n. The vector $\boldsymbol{\xi} = (\xi, \eta)$ is the vector of spatial coordinates in the reference system, while $\mathbf{x} = (x,y)$ is the spatial coordinate vector in the physical system. The spatial reference element T_e is the unit triangle composed of the nodes $\boldsymbol{\xi}_{e,1} = (\xi_{e,1}, \eta_{e,1}) = (0,0)$, $\boldsymbol{\xi}_{e,2} = (\xi_{e,2}, \eta_{e,2}) = (1,0)$ and $\boldsymbol{\xi}_{e,3} = (\xi_{e,2}, \eta_{e,2}) = (0,1)$.

As usual for finite volume schemes, data are represented by spatial cell averages, which are defined at time t^n as

$$\mathbf{Q}_i^n = \frac{1}{|T_i^n|} \int_{T_i^n} \mathbf{Q}(x,y,t^n) dV, \qquad (2.4)$$

where $|T_i^n|$ denotes the volume of element T_i^n at the current time t^n. To achieve higher order in space, piecewise high order polynomials $\mathbf{w}_h(x,y,t^n)$ must be reconstructed from the given cell averages (2.4) using a polynomial WENO reconstruction procedure illustrated briefly in the next section.

2.1 Polynomial WENO Reconstruction on Unstructured Meshes

In this paper we use the WENO reconstruction algorithm in the *polynomial* formulation presented in [26–28], instead of adopting the original *pointwise* WENO scheme [45,47,80]. Here we will give a very brief summary of the algorithm, since it is completely described in all detail in the above-mentioned references.

The reconstruction is performed by using the *reference system* (ξ,η) according to the mapping (2.3), as also explained in detail in [27]. The reconstruction polynomial of degree M is obtained by considering a reconstruction stencil S_i^s

$$\mathcal{S}_i^s = \bigcup_{j=1}^{n_e} T_{m(j)}^n.$$

(2.5)

The stencil contains a total number of elements n_e that is greater than the smallest number $\mathcal{M} = (M+1)(M+2)/2$ needed to reach the formal order of accuracy $M+1$, as shown in [4,49,61]. Typically we take $n_e = 2\mathcal{M}$ in two space dimensions. Here $1 \le j \le n_e$ is a local index counting the elements belonging to the stencil, while $m(j)$ maps the local index to the global element numbers used in the triangulation (2.2). s denotes the number and the position of the stencil with respect to the central element T_i^n. According to [27, 28] we always will use one central reconstruction stencil given by $s = 0$, three primary sector stencils and three reverse sector stencil, as suggested by Käser and Iske [49], so that we globally deal with seven stencils per element.

We use some spatial basis functions $\psi_l(\xi,\eta)$ in order to write the reconstruction polynomial for each candidate stencil s for triangle T_i^n:

$$\mathbf{w}_h^s(x,y,t^n) = \sum_{l=1}^{\mathcal{M}} \psi_l(\xi,\eta)\hat{\mathbf{w}}_{l,i}^{n,s} := \psi_l(\xi,\eta)\hat{\mathbf{w}}_{l,i}^{n,s},$$

(2.6)

with the mapping to the reference coordinate system given by the transformation (2.3). In the rest of the paper we will use classical tensor index notation with the Einstein summation convention, which implies summation over two equal indices. The number of the unknown polynomial coefficients (degrees of freedom) is $\mathcal{M}$, already defined previously. As basis functions $\psi_l(\xi,\eta)$ we use the orthogonal Dubiner–type basis on the reference triangle T_e, given in detail in [19,24,48].

Integral conservation is required for the reconstruction on each element $T_j^n \in \mathcal{S}_i^s$, hence

$$\frac{1}{|T_j^n|} \int_{T_j^n} \psi_l(\xi,\eta)\hat{\mathbf{w}}_{l,i}^{n,s}dV = \mathbf{Q}_j^n, \qquad \forall T_j^n \in \mathcal{S}_i^s$$

(2.7)

where $|T_j^n|$ represents the volume of element T_j^n at time t^n. Since the number of stencil elements is *larger* than the one of the unknown polynomial coefficients ($n_e > \mathcal{M}$), the above system given by eqn. (2.7) is an overdetermined linear algebraic system that is

readily solved for the unknown coefficients $\hat{\mathbf{w}}_{l,i}^{n,s}$ using a constrained least–squares technique, see [27]. The multi–dimensional integrals are evaluated using Gaussian quadrature formulae of suitable order, see the book of Stroud for details [69]. Since the shape of the triangles changes in time, the small linear systems given by (2.7) must be solved at the beginning of each time step, while the choice of the stencils $\mathcal{S}_i^s$ remains *fixed* for all times. It is therefore very useful to devise a one–step time–discretization, such as the one detailed in the next section, which requires only one reconstruction per time step, in contrast to higher order Runge–Kutta methods, which would require the evaluation of the above reconstruction operator in each substage of the Runge–Kutta method.

In order to obtain essentially non–oscillatory properties the final WENO reconstruction polynomial is computed from the reconstruction polynomials obtained on each individual stencil $\mathcal{S}_i^s$ in the usual way. For this purpose we adopt the classical definitions of the oscillation indicators σ_s reported in [47] and the oscillation indicator matrix Σ_{lm} proposed in [27, 28], which read

$$\sigma_s = \Sigma_{lm} \hat{w}_{l,i}^{n,s} \hat{w}_{m,i}^{n,s}, \qquad \Sigma_{lm} = \sum_{\alpha+\beta \leq M} \int_{T_e} \frac{\partial^{\alpha+\beta} \psi_l(\xi,\eta)}{\partial \xi^\alpha \partial \eta^\beta} \cdot \frac{\partial^{\alpha+\beta} \psi_m(\xi,\eta)}{\partial \xi^\alpha \partial \eta^\beta} d\xi d\eta. \qquad (2.8)$$

The nonlinear weights ω_s are defined by

$$\tilde{\omega}_s = \frac{\lambda_s}{(\sigma_s + \epsilon)^r}, \qquad \omega_s = \frac{\tilde{\omega}_s}{\sum_q \tilde{\omega}_q}, \qquad (2.9)$$

where we use $\epsilon = 10^{-14}$, $r = 8$, $\lambda_s = 1$ for the one–sided stencils and $\lambda_0 = 10^5$ for the central stencil, according to [26, 28]. The final nonlinear WENO reconstruction polynomial and its coefficients are then given by

$$\mathbf{w}_h(x,y,t^n) = \sum_{l=1}^{\mathcal{M}} \psi_l(\xi,\eta) \hat{\mathbf{w}}_{l,i}^n, \qquad \text{with} \qquad \hat{\mathbf{w}}_{l,i}^n = \sum_s \omega_s \hat{\mathbf{w}}_{l,i}^{n,s}. \qquad (2.10)$$

2.2 Local Space–Time Continuous–Galerkin Predictor on Moving Meshes

In order to obtain high order accuracy in time, the reconstructed polynomials $\mathbf{w}_h$ obtained at the current time t^n are now evolved *locally* within each element $T_i(t)$ during one time step $t^n \leq t \leq t^{n+1}$. The result of this local evolution step is an *element–local* piecewise polynomial space–time predictor $\mathbf{q}_h(x,y,t)$. For this purpose we use an element–local weak formulation of the PDE in both space and time, based on a Lagrangian version of the local space–time continuous Galerkin method introduced for the Eulerian framework in [25].

Let $\tilde{\mathbf{x}} = (x,y,t)$ denote the physical coordinate vector and $\tilde{\boldsymbol{\xi}} = (\xi,\eta,\tau)$ the reference coordinate vector, in both of which also the time is included, while $\mathbf{x} = (x,y)$ and $\boldsymbol{\xi} = (\xi,\eta)$ are the pure spatial coordinate vectors in physical and reference coordinates, respectively.

Let furthermore $\theta_l = \theta_l(\tilde{\boldsymbol{\xi}}) = \theta_l(\xi,\eta,\tau)$ be a space–time basis function defined by the Lagrange interpolation polynomials passing through the space–time nodes $\tilde{\boldsymbol{\xi}}_m = (\xi_m,\eta_m,\tau_m)$. The nodes are specified according to [25]. Being the Lagrange interpolation polynomials, the resulting *nodal* basis functions θ_l therefore satisfy the interpolation property

$$\theta_l(\tilde{\boldsymbol{\xi}}_m) = \delta_{lm}, \tag{2.11}$$

where δ_{lm} is the usual Kronecker symbol. According to [25] the local space-time solution in element $T_i(t)$, denoted by $\mathbf{q}_h$, as well as the fluxes $(\mathbf{f}_h, \mathbf{g}_h)$ and the source term $\mathbf{S}_h$ are approximated as

$$\mathbf{q}_h = \mathbf{q}_h(\xi,\eta,\tau) = \theta_l(\xi,\eta,\tau)\widehat{\mathbf{q}}_{l,i}, \qquad \mathbf{S}_h = \mathbf{S}_h(\xi,\eta,\tau) = \theta_l(\xi,\eta,\tau)\widehat{\mathbf{S}}_{l,i},$$
$$\mathbf{f}_h = \mathbf{f}_h(\xi,\eta,\tau) = \theta_l(\xi,\eta,\tau)\widehat{\mathbf{f}}_{l,i}, \qquad \mathbf{g}_h = \mathbf{g}_h(\xi,\eta,\tau) = \theta_l(\xi,\eta,\tau)\widehat{\mathbf{g}}_{l,i}. \tag{2.12}$$

The mapping between the physical space–time coordinate vector $\tilde{\mathbf{x}}$ and the reference space–time coordinate vector $\tilde{\boldsymbol{\xi}}$ is achieved using an *isoparametric* approach, i.e. the mapping is represented by the *same* basis functions θ_l as the approximate numerical solution itself, given in Eqn. (2.12) above. Hence, one has

$$x(\xi,\eta,\tau) = \theta_l(\xi,\eta,\tau)\widehat{x}_{l,i}, \qquad y(\xi,\eta,\tau) = \theta_l(\xi,\eta,\tau)\widehat{y}_{l,i}, \qquad t(\xi,\eta,\tau) = \theta_l(\xi,\eta,\tau)\widehat{t}_l, \tag{2.13}$$

where the degrees of freedom $\widehat{\mathbf{x}}_{l,i} = (\widehat{x}_{l,i}, \widehat{y}_{l,i})$ denote the (in parts unknown) vector of physical coordinates in space of the moving space–time control volume and the $\widehat{t}_l$ denote the *known* degrees of freedom of the physical time at each space–time node $\tilde{\mathbf{x}}_{l,i} = (\widehat{x}_{l,i}, \widehat{y}_{l,i}, \widehat{t}_l)$. The mapping in time simply reads

$$t = t_n + \tau\Delta t, \qquad \tau = \frac{t - t^n}{\Delta t}, \qquad \Rightarrow \qquad \widehat{t}_l = t_n + \tau_l\Delta t, \tag{2.14}$$

where t^n is the current time and Δt is the time step. Hence, $t_\xi = t_\eta = 0$ and $t_\tau = \Delta t$.

The spatial mapping given in (2.13) and the temporal mapping (2.14) allow us to transform the physical space–time element to the unit reference space–time element $T_e \times [0,1]$. The Jacobian of the spatial and temporal transformation is given by

$$J_{st} = \frac{\partial\tilde{\mathbf{x}}}{\partial\tilde{\boldsymbol{\xi}}} = \begin{pmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ 0 & 0 & \Delta_t \end{pmatrix}, \tag{2.15}$$

and its inverse reads

$$J_{st}^{-1} = \frac{\partial\tilde{\boldsymbol{\xi}}}{\partial\tilde{\mathbf{x}}} = \begin{pmatrix} \xi_x & \xi_y & \xi_t \\ \eta_x & \eta_y & \eta_t \\ 0 & 0 & \frac{1}{\Delta t} \end{pmatrix}. \tag{2.16}$$

The local reference system and the inverse of the associated Jacobian matrix (2.16) are used to rewrite the governing PDE (2.1) as

$$\frac{\partial\mathbf{Q}}{\partial\tau}\tau_t + \frac{\partial\mathbf{Q}}{\partial\xi}\xi_t + \frac{\partial\mathbf{Q}}{\partial\eta}\eta_t + \frac{\partial\mathbf{f}}{\partial\tau}\tau_x + \frac{\partial\mathbf{f}}{\partial\xi}\xi_x + \frac{\partial\mathbf{f}}{\partial\eta}\eta_x + \frac{\partial\mathbf{g}}{\partial\tau}\tau_y + \frac{\partial\mathbf{g}}{\partial\xi}\xi_y + \frac{\partial\mathbf{g}}{\partial\eta}\eta_y = \mathbf{S}(\mathbf{Q}), \tag{2.17}$$

which simplifies to

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \Delta t \left[\frac{\partial \mathbf{Q}}{\partial \xi} \xi_t + \frac{\partial \mathbf{Q}}{\partial \eta} \eta_t + \frac{\partial \mathbf{f}}{\partial \xi} \xi_x + \frac{\partial \mathbf{f}}{\partial \eta} \eta_x + \frac{\partial \mathbf{g}}{\partial \xi} \xi_y + \frac{\partial \mathbf{g}}{\partial \eta} \eta_y \right] = \Delta t \mathbf{S}(\mathbf{Q}), \qquad (2.18)$$

since $\tau_x = \tau_y = 0$ and $\tau_t = \frac{1}{\Delta t}$, according to the definition (2.14).

To make the notation easier we now introduce the following two integral operators

$$[f,g]^\tau = \int_{T_e} f(\xi,\eta,\tau) g(\xi,\eta,\tau) d\xi d\eta, \quad \langle f,g \rangle = \int_0^1 \int_{T_e} f(\xi,\eta,\tau) g(\xi,\eta,\tau) d\xi d\eta d\tau, \qquad (2.19)$$

which denote the scalar products of two functions f and g over the spatial reference element T_e at time τ and over the space-time reference element $T_e \times [0,1]$, respectively.

Inserting the definitions of (2.12) into the weak formulation of the PDE (2.18), we multiply Eqn. (2.18) with the same space–time basis functions $\theta_k(\xi,\eta,\tau)$ and then integrate it over the space–time reference element $T_e \times [0,1]$, hence obtaining:

$$\left\langle \theta_k, \frac{\partial \theta_l}{\partial \tau} \right\rangle \widehat{\mathbf{q}}_{l,i} \quad + \quad \Delta t \left(\left\langle \theta_k, \frac{\partial \theta_l}{\partial \xi} \xi_t \right\rangle + \left\langle \theta_k, \frac{\partial \theta_l}{\partial \eta} \eta_t \right\rangle \right) \widehat{\mathbf{q}}_{l,i}$$

$$+ \quad \Delta t \left(\left\langle \theta_k, \frac{\partial \theta_l}{\partial \xi} \xi_x \right\rangle + \left\langle \theta_k, \frac{\partial \theta_l}{\partial \eta} \eta_x \right\rangle \right) \widehat{\mathbf{f}}_{l,i}$$

$$+ \quad \Delta t \left(\left\langle \theta_k, \frac{\partial \theta_l}{\partial \xi} \xi_y \right\rangle + \left\langle \theta_k, \frac{\partial \theta_l}{\partial \eta} \eta_y \right\rangle \right) \widehat{\mathbf{g}}_{l,i} = \langle \theta_k, \theta_l \rangle \widehat{\mathbf{S}}_{l,i}.$$

$$(2.20)$$

The above expression can be written in a more compact matrix form, which reads

$$\mathbf{K}_\tau \widehat{\mathbf{q}}_{l,i} + \Delta t \left(\mathbf{K}_t \widehat{\mathbf{q}}_{l,i} + \mathbf{K}_x \widehat{\mathbf{f}}_{l,i} + \mathbf{K}_y \widehat{\mathbf{g}}_{l,i} \right), = \Delta t \mathbf{M} \widehat{\mathbf{S}}_{l,i}, \qquad (2.21)$$

with the following matrix definitions:

$$\mathbf{K}_\tau = \left\langle \theta_k, \frac{\partial \theta_l}{\partial \tau} \right\rangle, \qquad \mathbf{M} = \langle \theta_k, \theta_l \rangle, \qquad (2.22)$$

and

$$\mathbf{K}_t = \left\langle \theta_k, \frac{\partial \theta_l}{\partial \xi} \xi_t \right\rangle + \left\langle \theta_k, \frac{\partial \theta_l}{\partial \eta} \eta_t \right\rangle,$$

$$\mathbf{K}_x = \left\langle \theta_k, \frac{\partial \theta_l}{\partial \xi} \xi_x \right\rangle + \left\langle \theta_k, \frac{\partial \theta_l}{\partial \eta} \eta_x \right\rangle,$$

$$\mathbf{K}_y = \left\langle \theta_k, \frac{\partial \theta_l}{\partial \xi} \xi_y \right\rangle + \left\langle \theta_k, \frac{\partial \theta_l}{\partial \eta} \eta_y \right\rangle. \qquad (2.23)$$

Note that the Lagrangian nature of the scheme, i.e. the moving space–time control volume, leads to the term $\mathbf{K}_t\widehat{\mathbf{q}}_h$, which is not present in the Eulerian case introduced in [25]. In (2.21), only the matrix $\mathbf{K}_\tau$ and the space–time mass matrix $\mathbf{M}$ given in (2.22) can be pre–computed and stored. The other integrals continuously change in time, since they depend on the inverse of the Jacobian matrix, which varies according to the Lagrangian mesh motion. Therefore it may be more convenient to assemble a unified term $\mathbf{P}$ as

$$\mathbf{P}:=\mathbf{S}(\mathbf{Q})-\left(\mathbf{Q}_\xi\xi_t+\mathbf{Q}_\eta\eta_t+\mathbf{f}_\xi\xi_x+\mathbf{f}_\eta\eta_x+\mathbf{g}_\xi\xi_y+\mathbf{g}_\eta\eta_y\right), \tag{2.24}$$

hence

$$\mathbf{Q}_\tau=\Delta t\mathbf{P}. \tag{2.25}$$

$\mathbf{P}$ is approximated again inside element $T_i(t)$ by adopting a nodal approach as done in (2.12), i.e.

$$\mathbf{P}_h=\mathbf{P}_h(\xi,\eta,\tau)=\theta_l(\xi,\eta,\tau)\widehat{\mathbf{P}}_{l,i}, \tag{2.26}$$

with $\widehat{\mathbf{P}}_{l,i}=\mathbf{P}(\tilde{\mathbf{x}}_{l,i})$.

Let us denote with $\widehat{\mathbf{q}}_{l,i}^0$ the degrees of freedom that are known from the initial condition $\mathbf{w}_h$ so that $\mathbf{q}_h(x,y,t^n)=\mathbf{w}_h(x,y,t^n)$, whereas the unknown degrees of freedom for $\tau>0$ are denoted by $\widehat{\mathbf{q}}_{l,i}^1$, so that the total vector of degrees of freedom is written as $\widehat{\mathbf{q}}_{l,i}=(\widehat{\mathbf{q}}_{l,i}^0,\widehat{\mathbf{q}}_{l,i}^1)$. One can move onto the right-hand side of (2.21) or (2.25) all the known degrees of freedom $\widehat{\mathbf{q}}_{l,i}^0$ given by the initial condition $\mathbf{w}_h$ by setting the corresponding degrees of freedom to the known values, like a standard Dirichlet boundary condition in the continuous finite element framework, see [25] for more details. The final expression for the iterative scheme solving the nonlinear algebraic equation system of the two-dimensional Lagrangian continuous Galerkin predictor method simply reads

$$\mathbf{K}_\tau\widehat{\mathbf{q}}_{l,i}^{r+1}=\Delta t\mathbf{M}\widehat{\mathbf{P}}_{l,i}^r, \tag{2.27}$$

where the superscript r denotes the iteration number here. For an efficient initial guess ($r=0$) based on a second order MUSCL–type scheme, see [43].

Since the mesh is moving we also have to consider the evolution of the vertex coordinates of the local space–time element, whose motion is described by the ODE system

$$\frac{d\mathbf{x}}{dt}=\mathbf{V}(x,y,t), \tag{2.28}$$

where $\mathbf{V}=\mathbf{V}(x,y,t)=(U,V)$ is the local mesh velocity. Since we are developing an *arbitrary Lagrangian-Eulerian* scheme (ALE), we want the mesh velocity to be independent from the fluid velocity. In this framework we can deal in the same way with both Eulerian and Lagrangian schemes: if $\mathbf{V}=0$ the scheme reduces indeed to a pure Eulerian approach, while if $\mathbf{V}$ coincides with the local fluid velocity $\mathbf{v}$ we obtain a Lagrangian–type method. The velocity inside element $T_i(t)$ can be also expressed as

$$\mathbf{V}_h=\mathbf{V}_h(\xi,\eta,\tau)=\theta_l(\xi,\eta,\tau)\widehat{\mathbf{V}}_{l,i}, \tag{2.29}$$

with $\widehat{\mathbf{V}}_{l,i} = \mathbf{V}(\tilde{\mathbf{x}}_{l,i})$.

As introduced in [30], we can solve the ODE (2.28) for the unknown coordinate vector $\widehat{\mathbf{x}}_l$ using again the local space–time CG method:

$$\left\langle \theta_k, \frac{\partial \theta_l}{\partial \tau} \right\rangle \widehat{\mathbf{x}}_{l,i} = \Delta t \left\langle \theta_k, \theta_l \right\rangle \widehat{\mathbf{V}}_{l,i}, \tag{2.30}$$

hence resulting in the following iteration scheme for the element–local space–time predictor for the nodal coordinates:

$$\mathbf{K}_\tau \widehat{\mathbf{x}}_{l,i}^{r+1} = \Delta t \mathbf{M} \widehat{\mathbf{V}}_{l,i}^r. \tag{2.31}$$

The nodal degrees of freedom $\widehat{\mathbf{x}}_l$ at relative time $\tau = 0$, i.e. the initial condition of the ODE system, are given by the spatial mapping (2.3), since the physical triangle T_i^n at time t^n is known.

Eqn. (2.30) is iterated *together* with the weak formulation for the solution (2.27). The algorithm for the two–dimensional local space–time CG predictor for moving meshes can be summarized by the following steps:

- compute the local mesh velocity (2.29), usually by choosing the fluid velocity, hence $\widehat{\mathbf{V}}_{l,i} = \mathbf{v}(\tilde{\mathbf{x}}_{l,i})$;

- with (2.31) update the geometry *locally* within the predictor stage, i.e. the element–local space–time coordinates;

- compute the Jacobian matrix and its inverse by using (2.15)-(2.16);

- compute the term $\mathbf{P}$ according to (2.25);

- evolve the solution *locally* with (2.27).

The iterative procedure described above stops when the residuals of (2.27) and (2.31) are less than a prescribed tolerance *tol* (typically $tol \approx 10^{-12}$).

Once we have carried out the above procedure for all the elements of the computational domain, we end up with an *element–local predictor* for the numerical solution $\mathbf{q}_h$, for the fluxes $\mathbf{F}_h = (\mathbf{f}_h, \mathbf{g}_h)$, for the source term $\mathbf{S}_h$ and also for the mesh velocity $\mathbf{V}_h$.

Next, we have to update the mesh *globally*. Let us denote with $\mathcal{V}_k$ the neighborhood of vertex number k, i.e. all those elements that have in common the node number k. The number of elements in the neighborhood $\mathcal{V}_k$ is denoted with N_k. Since the velocity of each vertex is defined by the local predictor within each element, one has to deal with several, in general different, velocities for the same node, since all elements belonging to $\mathcal{V}_k$ will in general give a different velocity contribution, according to their element–local predictor. Since we do not admit the geometry to be discontinuous, we decide to fix a

unique node velocity $\overline{\mathbf{V}}_k^n$ to move the node. The final velocity is chosen to be the *average velocity* considering all the contributions $\overline{\mathbf{V}}_{k,j}^n$ of the vertex neighborhood as

$$\overline{\mathbf{V}}_k^n = \frac{1}{N_k} \sum_{T_j^n \in \mathcal{V}_k} \overline{\mathbf{V}}_{k,j}^n, \qquad \text{with} \qquad \overline{\mathbf{V}}_{k,j}^n = \left(\int_0^1 \theta_l(\xi_{e,m(k)}, \eta_{e,m(k)}, \tau) d\tau \right) \widehat{\mathbf{V}}_{l,j}. \tag{2.32}$$

The $\xi_{e,m(k)}$ and $\eta_{e,m(k)}$ are the vertex coordinates of the reference triangle T_e corresponding to vertex number k, hence $m = m(k)$ with $1 \le m \le 3$ is a mapping from the global node number k to the element–local vertex number. Since each node now has its own unique velocity, the vertex coordinates can be moved according to

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t \overline{\mathbf{V}}_k^n, \tag{2.33}$$

and we can update all the other geometric quantities needed for the computation, e.g. normal vectors, volumes, side lengths, barycenter position, *etc.*

2.3 Finite Volume Scheme

The conservation law (2.1) can be easily rewritten in a more compact space–time divergence form, which reads

$$\tilde{\nabla} \cdot \tilde{\mathbf{Q}} = \mathbf{S}(\mathbf{Q}), \tag{2.34}$$

with

$$\tilde{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial t} \right)^T, \qquad \tilde{\mathbf{Q}} = (\mathbf{F}, \mathbf{Q}) = (\mathbf{f}, \mathbf{g}, \mathbf{Q}). \tag{2.35}$$

Integration over the space–time control volume $C_i^n = T_i(t) \times [t^n; t^{n+1}]$ yields

$$\int_{t^n}^{t^{n+1}} \int_{T_i(t)} \tilde{\nabla} \cdot \tilde{\mathbf{Q}} \, d\mathbf{x} dt = \int_{t^n}^{t^{n+1}} \int_{T_i(t)} \mathbf{S} \, d\mathbf{x} dt. \tag{2.36}$$

Applying Gauss' theorem allows us to write the left space–time volume integral above as sum of the flux integrals computed over the space–time surface ∂C_i^n, hence

$$\int_{\partial C_i^n} \tilde{\mathbf{Q}} \cdot \tilde{\mathbf{n}} \, dS = \int_{t^n}^{t^{n+1}} \int_{T_i(t)} \mathbf{S} \, d\mathbf{x} dt, \tag{2.37}$$

where $\tilde{\mathbf{n}} = (\tilde{n}_x, \tilde{n}_y, \tilde{n}_t)$ is the outward pointing space–time unit normal vector on the space–time surface ∂C_i^n.

The space–time surface ∂C_i^n above involves overall five space–time sub–surfaces, as depicted in Figure 1:

$$\partial C_i^n = \left(\bigcup_{T_j(t) \in \mathcal{N}_i} \partial C_{ij}^n \right) \cup T_i^n \cup T_i^{n+1}, \tag{2.38}$$

where $\mathcal{N}_i$ denotes the so–called *Neumann neighborhood* of triangle $T_i(t)$, i.e. the set of directly adjacent triangles $T_j(t)$ that share a common edge $\partial T_{ij}(t)$ with triangle $T_i(t)$. The common space–time edge ∂C_{ij}^n during the time interval $[t^n; t^{n+1}]$ is denoted above by $\partial C_{ij}^n = \partial T_{ij}(t) \times [t^n; t^{n+1}]$.

The upper space–time sub–surface T_i^{n+1} and the lower space–time sub–surface T_i^n are parametrized by $0 \leq \xi \leq 1 \wedge 0 \leq \eta \leq 1 - \xi$ and the mapping (2.3). They are orthogonal to the time coordinate, hence for these faces the space–time unit normal vectors simply read $\tilde{\mathbf{n}} = (0,0,1)$ for T_i^{n+1} and $\tilde{\mathbf{n}} = (0,0,-1)$ for T_i^n, respectively.

The lateral space–time sub–faces ∂C_{ij}^n are defined using a simple bilinear parametrization, since the old vertex coordinates $\mathbf{X}_{ik}^n$ are given and the new ones $\mathbf{X}_{ik}^{n+1}$ are known from (2.33).

$$\partial C_{ij}^n = \tilde{\mathbf{x}}(\chi,\tau) = \sum_{k=1}^{4} \beta_k(\chi,\tau) \tilde{\mathbf{X}}_{ij,k}^n, \qquad 0 \leq \chi \leq 1, \quad 0 \leq \tau \leq 1, \tag{2.39}$$

where (χ,τ) represents a side-aligned local reference system according to Figure 1. The $\tilde{\mathbf{X}}_{ij,k}^n$ are the physical space–time coordinate vectors for the four vertices that define the lateral space–time sub–surface ∂C_{ij}^n. If $\mathbf{X}_{ij,1}^n$ and $\mathbf{X}_{ij,2}^n$ denote the two spatial nodes at time t^n that define the common spatial edge $\partial T_{ij}(t^n)$, then the four vectors $\tilde{\mathbf{X}}_{ij,k}^n$ are given by

$$\tilde{\mathbf{X}}_{ij,1}^n = \left(\mathbf{X}_{ij,1}^n, t^n \right), \qquad \tilde{\mathbf{X}}_{ij,2}^n = \left(\mathbf{X}_{ij,2}^n, t^n \right), \qquad \tilde{\mathbf{X}}_{ij,3}^n = \left(\mathbf{X}_{ij,2}^{n+1}, t^{n+1} \right), \qquad \tilde{\mathbf{X}}_{ij,4}^n = \left(\mathbf{X}_{ij,1}^{n+1}, t^{n+1} \right). \tag{2.40}$$

The $\beta_k(\chi,\tau)$ are a set of bilinear basis functions, which are defined as

$$\begin{aligned} \beta_1(\chi,\tau) &= (1-\chi)(1-\tau), \\ \beta_2(\chi,\tau) &= \chi(1-\tau), \\ \beta_3(\chi,\tau) &= \chi\tau, \\ \beta_4(\chi,\tau) &= (1-\chi)\tau. \end{aligned} \tag{2.41}$$

From (2.40) and (2.41) it follows that the temporal mapping is again simply $t = t^n + \tau \Delta t$, hence $t_\chi = 0$ and $t_\tau = \Delta t$.

The determinant of the coordinate transformation and the resulting space–time unit normal vector $\tilde{\mathbf{n}}_{ij}$ of the sub–surface ∂C_{ij}^n can be computed as follows:

$$|\partial C_{ij}^n| = \left| \frac{\partial \tilde{\mathbf{x}}}{\partial \chi} \times \frac{\partial \tilde{\mathbf{x}}}{\partial \tau} \right|, \quad \tilde{\mathbf{n}}_{ij} = \left(\frac{\partial \tilde{\mathbf{x}}}{\partial \chi} \times \frac{\partial \tilde{\mathbf{x}}}{\partial \tau} \right) / |\partial C_{ij}^n|. \tag{2.42}$$

Therefore, the ALE–type one–step finite volume scheme takes the following form:

$$|T_i^{n+1}| \mathbf{Q}_i^{n+1} = |T_i^n| \mathbf{Q}_i^n - \sum_{T_j \in \mathcal{N}_i} \int_0^1 \int_0^1 |\partial C_{ij}^n| \tilde{\mathbf{Q}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} \, d\chi \, d\tau + \int_{t^n}^{t^{n+1}} \int_{T_i(t)} \mathbf{S}(\mathbf{q}_h) \, d\mathbf{x} \, dt, \tag{2.43}$$
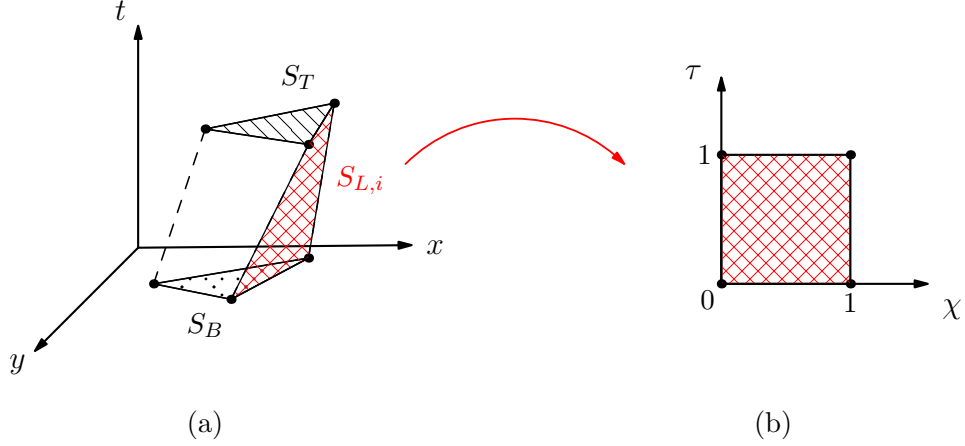
Figure 1: Physical space–time element (a) and parametrization of the lateral space–time subsurface ∂C_{ij}^n (b).

where $\tilde{\mathbf{Q}}_{ij} \cdot \tilde{\mathbf{n}}_{ij}$ is an Arbitrary–Lagrangian–Eulerian numerical flux function to resolve the discontinuity of the predictor solution $\mathbf{q}_h$ at the space–time sub–face ∂C_{ij}^n. The surface integrals appearing in (2.43) are approximated using multidimensional Gaussian quadrature rules, see [69] for details. At the interface ∂C_{ij}^n let us denote the local space–time predictor solution inside element $T_i(t)$ by $\mathbf{q}_h^-$ and the element–local predictor solution of the neighbor element $T_j(t)$ by $\mathbf{q}_h^+$, then a simple ALE–type Rusanov flux [30] is given by

$$\tilde{\mathbf{Q}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = \frac{1}{2} \left(\tilde{\mathbf{Q}}(\mathbf{q}_h^+) + \tilde{\mathbf{Q}}(\mathbf{q}_h^-) \right) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} s_{\max} \left(\mathbf{q}_h^+ - \mathbf{q}_h^- \right), \qquad (2.44)$$

where $s_{\max}$ is the maximum eigenvalue of the ALE Jacobian matrix in spatial normal direction,

$$\mathbf{A_n^V}(\mathbf{Q}) = \partial(\mathbf{F} \cdot \mathbf{n})/\partial \mathbf{Q} - (\mathbf{V} \cdot \mathbf{n}) \mathbf{I}, \qquad \text{with} \qquad \mathbf{n} = \frac{(\tilde{n}_x, \tilde{n}_y)^T}{\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2}}. \qquad (2.45)$$

Here, $\mathbf{V} \cdot \mathbf{n}$ is the local normal mesh velocity and $\mathbf{I}$ is the $\nu \times \nu$ identity matrix. It can be easily verified that

$$\mathbf{V} = \frac{1}{\Delta t} \begin{pmatrix} x_\tau \\ y_\tau \end{pmatrix}, \quad \tilde{\mathbf{n}}_{ij} = \begin{pmatrix} y_\chi \Delta t \\ -x_\chi \Delta t \\ x_\chi y_\tau - y_\chi x_\tau \end{pmatrix}, \quad \text{hence} \quad \mathbf{V} \cdot \mathbf{n} = -\frac{\tilde{n}_t}{\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2}}. \qquad (2.46)$$

A more sophisticated Osher–type ALE flux has been introduced in the Eulerian case in [29] and has been employed in the Lagrangian framework in [30]. It reads

$$\tilde{\mathbf{Q}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = \frac{1}{2} \left(\tilde{\mathbf{Q}}(\mathbf{q}_h^+) + \tilde{\mathbf{Q}}(\mathbf{q}_h^-) \right) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} \left(\int_0^1 |\mathbf{A_n^V}(\mathbf{\Psi}(s))| \, ds \right) \left(\mathbf{q}_h^+ - \mathbf{q}_h^- \right), \qquad (2.47)$$

with the straight–line segment path connecting the left and right state as follows,

$$\boldsymbol{\Psi}(s) = \mathbf{q}_h^- + s\left(\mathbf{q}_h^+ - \mathbf{q}_h^-\right), \qquad 0 \le s \le 1. \tag{2.48}$$

In (2.47) above, the usual definition of the matrix absolute value operator applies, i.e.

$$|\mathbf{A}| = \mathbf{R}|\boldsymbol{\Lambda}|\mathbf{R}^{-1}, \qquad |\boldsymbol{\Lambda}| = \mathrm{diag}(|\lambda_1|, |\lambda_2|, ..., |\lambda_\nu|), \tag{2.49}$$

with the right eigenvector matrix $\mathbf{R}$ and its inverse $\mathbf{R}^{-1}$. According to [29] the path integral appearing in (2.47) is approximated using Gaussian quadrature rules of sufficient accuracy.

3 Test Problems

In order to validate the two–dimensional ALE finite volume scheme presented in the previous section, we present some computational test problems in the following. We consider the Euler equations of compressible gas dynamics, which read:

$$\mathbf{Q}_t + \mathbf{f}_x + \mathbf{g}_y = \mathbf{S}(x, y, t) \tag{3.1}$$

with

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(\rho E + p) \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(\rho E + p) \end{pmatrix}. \tag{3.2}$$

Here, ρ denotes the fluid density , $\mathbf{v} = (u, v)$ is the fluid's velocity vector and ρE represents the total energy density. The vector of source terms is denoted by $\mathbf{S}$ and the fluid pressure by p, given in terms of the conserved quantities by the equation of state of an ideal gas as

$$p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho(u^2 + v^2)\right), \tag{3.3}$$

with the ratio of specific heats γ.

For each of the following test cases we choose the local mesh velocity as the local fluid velocity according to Eqn. (2.32), i.e.

$$\mathbf{V} = \mathbf{v}. \tag{3.4}$$

3.1 Numerical Convergence Results

The numerical convergence studies for the two–dimensional Lagrangian finite volume scheme are carried out considering a smooth convected isentropic vortex, see [45]. The initial condition is given in terms of primitive variables and it consists in a linear superposition of a homogeneous background field and some perturbations δ:

$$(\rho, u, v, p) = (1 + \delta\rho, 1 + \delta u, 1 + \delta v, 1 + \delta p). \tag{3.5}$$

Table 1: Numerical convergence results for the compressible Euler equations using the first up to sixth order version of the two–dimensional Lagrangian one–step WENO finite volume schemes presented in this article. The error norms refer to the variable ρ (density) at time $t=1.0$.

$h(\Omega(t_f))$	ϵ_{L_2}	$\mathcal{O}(L_2)$	$h(\Omega,t_f)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	$h(\Omega,t_f)$	ϵ_{L_2}	$\mathcal{O}(L_2)$
	$\mathcal{O}1$			$\mathcal{O}2$			$\mathcal{O}3$	
3.73E-01	9.525E-02	-	3.43E-01	1.716E-02	-	3.28E-01	1.614E-02	-
2.63E-01	6.907E-02	0.9	2.49E-01	1.109E-02	1.4	2.51E-01	6.943E-03	3.0
2.14E-01	5.700E-02	0.9	1.69E-01	5.766E-03	1.7	1.68E-01	2.290E-03	2.7
1.74E-01	4.752E-02	0.9	1.28E-01	3.027E-03	2.3	1.28E-01	9.274E-04	3.3
	$\mathcal{O}4$			$\mathcal{O}5$			$\mathcal{O}6$	
3.29E-01	4.717E-03	-	3.29E-01	4.9463E-03	-	3.29E-01	2.051E-03	-
2.51E-01	1.822E-03	3.5	2.51E-01	1.4648E-03	4.5	2.51E-01	5.803E-04	4.7
1.67E-01	4.379E-04	3.5	1.67E-01	2.5937E-04	4.3	1.67E-01	8.317E-05	4.8
1.28E-01	1.313E-04	4.4	1.28E-01	6.9664E-05	4.9	1.31E-01	1.994E-05	5.9

We set the vortex radius $r^2 = (x-5)^2 + (y-5)^2$, the vortex strength $\epsilon = 5$ and the ratio of specific heats $\gamma = 1.4$. The perturbation of entropy $S = \frac{p}{\rho^\gamma}$ is assumed to be zero, while the perturbations of temperature T and velocity $\mathbf{v}$ are given by

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \frac{\epsilon}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y-5) \\ (x-5) \end{pmatrix}, \quad \delta S = 0, \quad \delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}. \tag{3.6}$$

From (3.6) it follows that the perturbations for density and pressure are given by

$$\delta\rho = (1+\delta T)^{\frac{1}{\gamma-1}} - 1, \quad \delta p = (1+\delta T)^{\frac{\gamma}{\gamma-1}} - 1. \tag{3.7}$$

The initial computational domain $\Omega(0) = [0;10] \times [0;10]$ is square–shaped and it is surrounded by four periodic boundary conditions. The exact solution $\mathbf{Q}_e$ is the time–shifted initial condition given by $\mathbf{Q}_e(\mathbf{x},t) = \mathbf{Q}(\mathbf{x}-\mathbf{v}_c t,0)$, with the convective mean velocity $\mathbf{v}_c = (1,1)$. We use the Osher–type numerical flux (2.47) to run this test case until a final time of $t_f = 1.0$ on successive refined meshes and for each mesh the corresponding error in L_2 norm is computed as

$$\epsilon_{L_2} = \sqrt{\int_{\Omega(t_f)} \left(\mathbf{Q}_e(x,y,t_f) - \mathbf{w}_h(x,y,t_f) \right)^2 dx dy}, \tag{3.8}$$

while the mesh size $h(\Omega(t_f))$ is taken to be the maximum diameter of the circumcircles of the triangles in the final domain $\Omega(t_f)$. The resulting numerical convergence rates are listed in Table 1, whereas Figure 2 shows the temporal evolution of three different meshes used for the numerical convergence study.
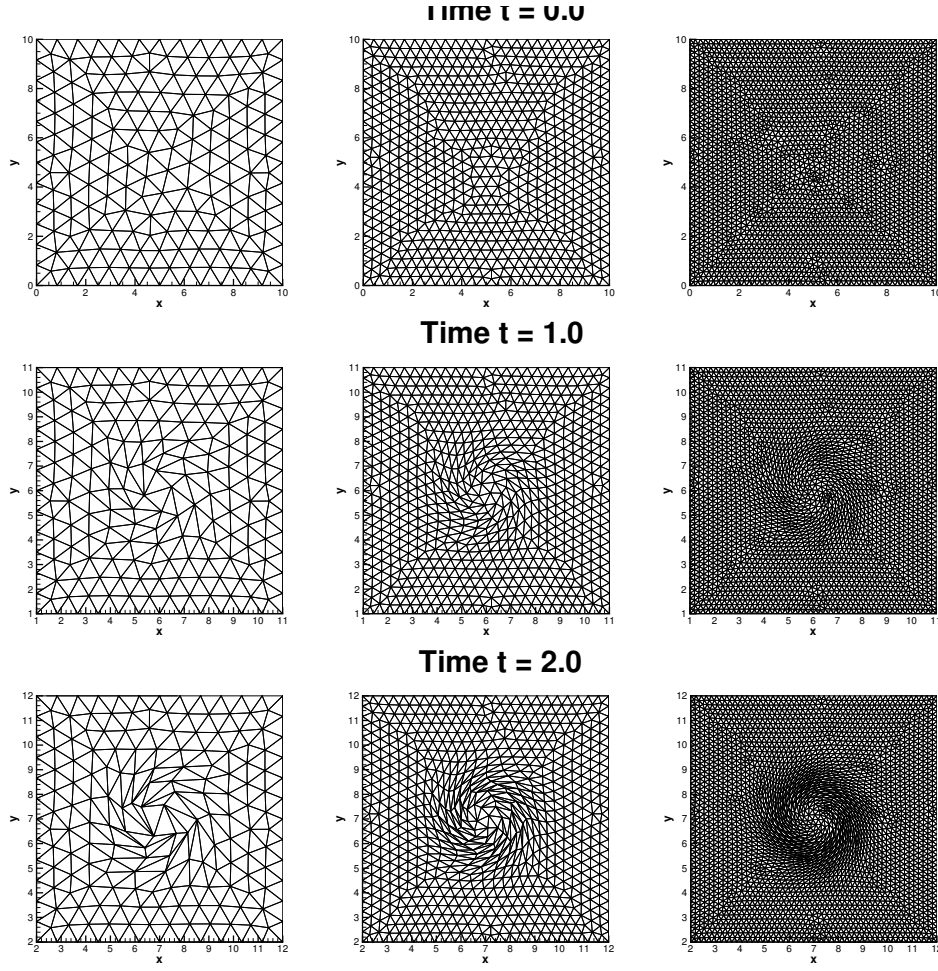
Figure 2: Different mesh sizes used for the convergence rates study at different time outputs: $t = 0$ (top row), $t = 1$ (middle row) and $t = 2$ (bottom row). The total number of elements N_E is increasing from the left grid ($N_E = 320$), passing through the middle one ($N_E = 1298$), to the right one ($N_E = 5180$).

3.2 Numerical Flux Comparison

In order to study how the choice of the numerical flux does affect the solution, we consider the well-known Sod shock tube problem. The rectangular shaped computational domain $\Omega(0) = [-0.5;0.5] \times [-0.1;0.1]$ contains a total number of elements of $N_E = 18018$. The initial condition reads

$$\mathbf{Q}(\mathbf{x},0) = \begin{cases} \mathbf{Q}_L = & (1.0,0,0,1.0) & \text{if} & \mathbf{x} \leq \mathbf{x}_0 \\ \mathbf{Q}_R = & (0.125,0,0,0.1) & \text{if} & \mathbf{x} > \mathbf{x}_0 \end{cases} \tag{3.9}$$

with the position vector $\mathbf{x}_0 = (0,y)$ which denotes the location of the discontinuity.

The Sod problem involves physically a rarefaction wave traveling towards left and both a contact and a shock wave which are moving to the right side of the domain. Exact solution is known from the usage of an exact Riemann solver, while the numerical results plotted in Figure 3 have been collected with three different numerical fluxes, namely the Rusanov–type and the Osher–type flux given by (2.47) and (2.44), respectively. Furthermore we use also an HLLC–type numerical flux, firstly introduced for moving meshes by Van der Vegt et al. [77,78] in the DG finite element framework. For a detailed description of the HLLC flux and its extension to dynamic grid motion we refer to [77].

Figure 3 shows a comparison between the exact solution and the numerical results at time $t = 0.25$ obtained with a third order accurate scheme. The Rusanov flux is more diffusive if compared with the Osher and the HLLC fluxes, especially looking at the contact wave located at $x \approx 0.23$, which tends to be smoothed, while on the contrary it is almost perfectly resolved by the Osher– and the HLLC–ALE scheme.

Convergence rate studies for the three different numerical fluxes are reported in Table 2. As test problem we use again the isentropic vortex fully described in section 3.1, but here we consider only a numerical scheme with an order of accuracy of $\mathcal{O}3$. The lowest L_2-norm error is achieved by the very accurate Osher–type flux, as well as for the HLLC flux, whose error does not differ so much from the Osher scheme. The Rusanov type gives the highest error, as can be also observed in Figure 3.

Table 2: Comparison of numerical convergence results for the compressible Euler equations using the third order version of the two–dimensional Lagrangian one–step WENO finite volume schemes and three different types of numerical fluxes (Rusanov, Osher and HLLC). The error norms refer to the variable ρ (density) at time $t = 1.0$.

Rusanov			Osher			HLLC		
$h(\Omega(t_f))$	ϵ_{L_2}	$\mathcal{O}(L_2)$	$h(\Omega,t_f)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	$h(\Omega,t_f)$	ϵ_{L_2}	$\mathcal{O}(L_2)$
3.61E-01	1.076E-01	-	3.28E-01	1.614E-02	-	3.31E-01	1.818E-02	-
2.51E-01	2.315E-02	4.2	2.51E-01	6.943E-03	3.2	2.51E-01	7.897E-03	3.0
1.68E-01	8.658E-03	2.4	1.68E-01	2.290E-03	2.7	1.68E-01	2.621E-03	2.7
1.28E-01	3.950E-03	2.9	1.28E-01	9.274E-04	3.3	1.28E-01	1.068E-03	3.3

Looking at the computational time in Table 3, the Rusanov flux is the most efficent

scheme, but also the less accurate, while the Osher–type version gives the lowest error and is not the most expensive one, since the HLLC flux requires normally major computational efforts. We take Rusanov computational time as reference, hence η evaluates simply the ratio between the current flux CPU time and the Rusanov one. Using HLLC one gets up to $\eta = 1.7$, while at most $\eta = 1.5$ is achieved with the Osher–type flux, which ensures the scheme to be the most accurate.

Table 3: Computational time for the convergence studies and the Sod shock tube results obtained with three different numerical fluxes (Rusanov, Osher and HLLC). CPU Time is measured in seconds $[s]$ and η denotes the ratio w.r.t. the Rusanov computational time.

	Rusanov	Osher		HLLC	
	CPU time	CPU time	η	CPU time	η
	6.21E+00	9.11E+00	1.5	1.06E+01	1.7
Isentropic	2.22E+01	2.49E+01	1.1	2.77E+01	1.2
vortex	9.64E+01	1.02E+02	1.1	1.12E+02	1.1
	1.55E+02	1.69E+02	1.1	1.93E+02	1.2
Sod problem	1.22E+04	1.44E+04	1.2	1.59E+04	1.3
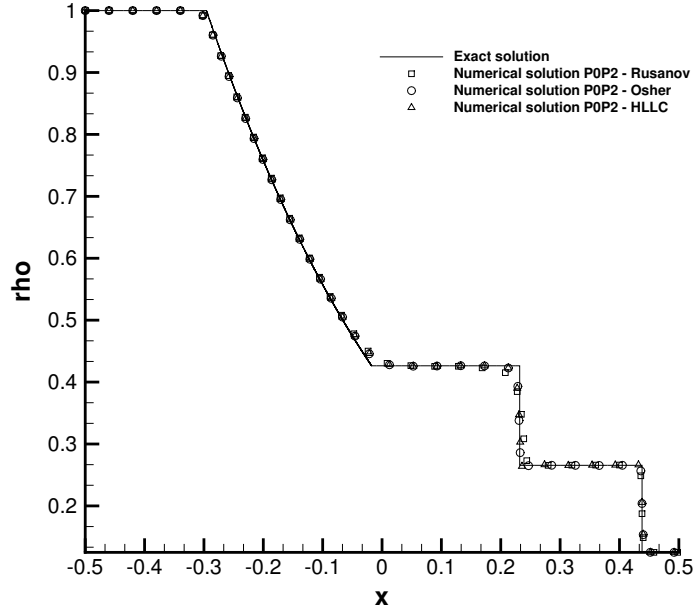
Figure 3: Sod shock tube test. Third order accurate numerical results obtained using three different numerical fluxes (Rusanov, Osher and HLLC) and comparison with the exact solution (solid line) at time $t = 0.25$.

3.3 A Two–Dimensional Explosion Problem

This test problem is in some sense a two–dimensional extension of the classical one–dimensional shock tube problems. The initial domain $\Omega(0)$ is a circle of radius $R_o = 1$ and the initial condition is given by

$$\mathbf{Q}(\mathbf{x},0) = \begin{cases} \mathbf{Q}_i & \text{if} \quad r \leq R, \\ \mathbf{Q}_o & \text{if} \quad r > R, \end{cases} \tag{3.10}$$

with $r^2 = x^2 + y^2$. The circle of radius $R = 0.5$ separates the *inner state* $\mathbf{Q}_i$ from the *outer state* $\mathbf{Q}_o$. We assume $\gamma = 1.4$ and use the initial condition reported in Table 4.

Table 4: Initial condition for the explosion test

	ρ	u	v	p
Inner state ($\mathbf{Q}_i$)	1.0	0.0	0.0	1.0
Outer state ($\mathbf{Q}_o$)	0.125	0.0	0.0	0.1

In order to compute a reliable reference solution we assume rotational symmetry. Hence, one can simplify the multidimensional Euler equations to a one–dimensional system with geometric source terms, as proposed in [75], which reads

$$\mathbf{Q}_t + \mathbf{F}(\mathbf{Q})_r = \mathbf{S}(\mathbf{Q}), \tag{3.11}$$

where

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(\rho E + p) \end{pmatrix}, \quad \mathbf{S} = -\frac{\alpha}{r} \begin{pmatrix} \rho u \\ \rho u^2 \\ u(\rho E + p) \end{pmatrix}. \tag{3.12}$$

Here, r denotes the radial direction and u is the radial velocity, while α is a parameter that allows the system (3.11) to be equivalent to the one–, two– or three–dimensional Euler equations with rotational symmetry, according to its value:

- $\alpha = 0$: plain 1D flow,

- $\alpha = 1$: cylindrical symmetry (2D flow),

- $\alpha = 2$: spherical symmetry (3D flow).

We choose $\alpha = 1$ for the 2D case and the inhomogeneous system of equations (3.11) is solved using a second order MUSCL scheme with the Rusanov flux on a one–dimensional mesh of 15000 points in the radial interval $r \in [0;1]$. This solution is assumed to be our reference solution, which will be used to verify the accuracy of the two–dimensional Lagrangian finite volume scheme. Figure 4 shows the comparison between the reference solution and the numerical solution obtained with the third order version of our Lagrangian one–step WENO scheme for density and velocity at time $t = 0.20$: a circular shock wave is travelling away from the center together with a contact wave, while a

circular rarefaction wave is running towards the origin. The contact wave is very well resolved due to the use of the little diffusive Osher–type flux. The mesh used contained 68,324 triangles of characteristic mesh spacing $h = 1/100$. A 3D view of the numerical solution as well as a very coarse version of the mesh are depicted in Fig. 5.
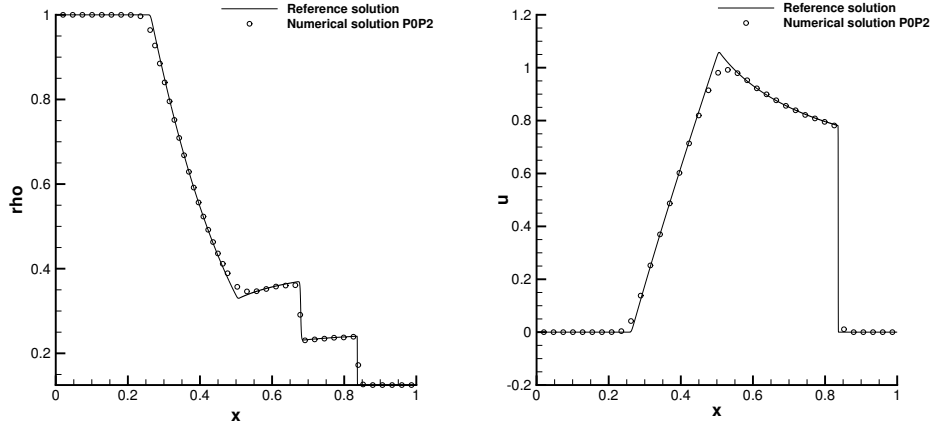

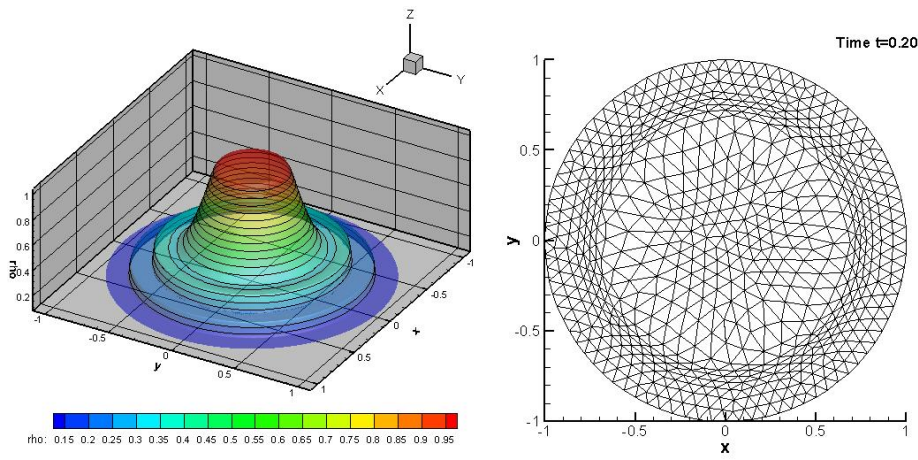Figure 4: Profiles along the positive x-axis of density (left) and velocity (right) at time $t = 0.2$.

Figure 5: Left: density distribution at $t = 0.20$ for the explosion test. Right: coarse distorted mesh at $t = 0.20$.

3.4 The Kidder Problem

The Kidder problem consists in an isentropic compression of a shell filled with an ideal gas. For this problem, an exact analytical solution has been proposed by Kidder in [50].

This is a very useful and widely used test [10, 56], to assure that a Lagrangian scheme does not produce spurious entropy. The shell has a time–dependent internal radius $r_i(t)$ and an external radius $r_e(t)$, while r denotes the general radial coordinate with $r_i(t) \leq r \leq r_e(t)$. The initial values for the internal and external radius are $r_i(0) = r_{i,0} = 0.9$ and $r_e(0) = r_{e,0} = 1.0$, respectively. In this test problem the ratio of specific heats is $\gamma = 2$ and the initial density distribution is given by

$$\rho_0 = \rho(r,0) = \left(\frac{r_{e,0}^2 - r^2}{r_{e,0}^2 - r_{i,0}^2} \rho_{i,0}^{\gamma-1} + \frac{r^2 - r_{i,0}^2}{r_{e,0}^2 - r_{e,0}^2} \rho_{e,0}^{\gamma-1} \right)^{\frac{1}{\gamma-1}} \tag{3.13}$$

with $\rho_{i,0} = 1$ and $\rho_{e,0} = 2$, which are the initial values of density defined at the internal and at the external frontier of the shell, respectively. The initial entropy $s_0 = \frac{p_0}{\rho_0^\gamma} = 1$ is uniform, so that the initial pressure distribution is given by

$$p_0(r) = s_0 \rho_0(r)^\gamma. \tag{3.14}$$

Initially the fluid is at rest, hence $u = v = 0$.

According to [50] we look for a solution of the the form $R(r,t) = h(t)r$, where $R(r,t)$ denotes the radius at time $t > 0$ of a fluid particle initially located at radius r. Therefore the self-similar analytical solution for $t \in [0, \tau]$ reads

$$\begin{aligned}
\rho(R(r,t),t) &= h(t)^{-\frac{2}{\gamma-1}} \rho_0 \left[\frac{R(r,t)}{h(t)} \right], \\
u_r(R(r,t),t) &= \frac{d}{dt} h(t) \left[\frac{R(r,t)}{h(t)} \right], \\
p(R(r,t),t) &= h(t)^{-\frac{2\gamma}{\gamma-1}} p_0 \left[\frac{R(r,t)}{h(t)} \right],
\end{aligned} \tag{3.15}$$

where the homothety rate is

$$h(t) = \sqrt{1 - \frac{t^2}{\tau^2}}, \tag{3.16}$$

with τ representing the focalisation time

$$\tau = \sqrt{\frac{\gamma-1}{2} \frac{(r_{e,0}^2 - r_{i,0}^2)}{c_{e,0}^2 - c_{i,0}^2}}, \tag{3.17}$$

and the internal and external sound speeds c_i and c_e are defined as

$$c_i = \sqrt{\gamma \frac{p_i}{\rho_i}}, \qquad c_e = \sqrt{\gamma \frac{p_e}{\rho_e}}. \tag{3.18}$$

The boundary conditions are imposed at the internal and the external frontier of the shell, where a space–time dependent state is prescribed according to the exact analytical solution (3.15).

As done by Carré et al. in [10] the final time of the simulation is chosen to be $t_f = \frac{\sqrt{3}}{2}\tau$, so that the resulting compression rate is $h(t_f) = 0.5$. In this way the exact solution is given by a shell bounded with $0.45 \leq R \leq 0.5$.
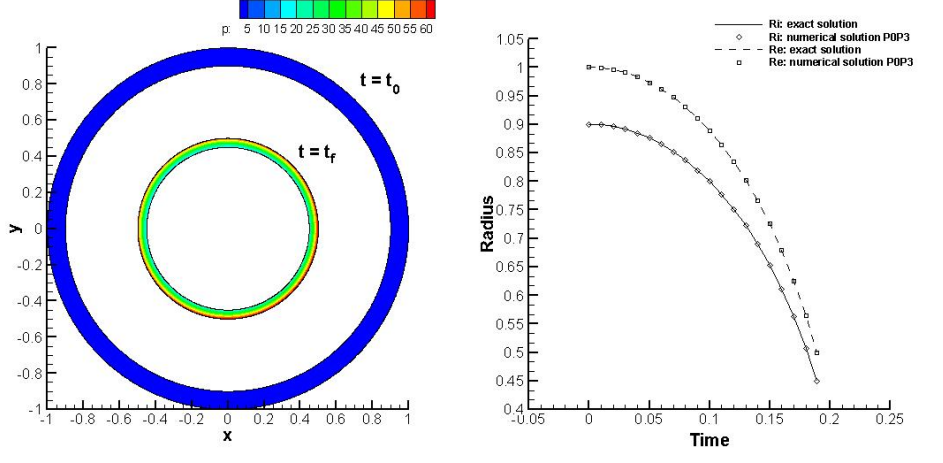

Figure 6: Left: Numerical solution at $t = 0.4$ and at $t = t_f$. Right: Evolution of the internal and external radius of the shell and comparison between analytical and numerical solution.

We use the Osher–type numerical flux (2.47) to obtain the results shown in Figure 6, where we can observe an excellent agreement between the analytical and the numerical solution. The absolute error $|err|$ concerning the location of the internal and external radius at the end of the simulation has also be computed and is reported in Table 5.

	r_{ex}	r_{num}	$|err|$
Internal radius	0.4500000	0.4499936	6.40E-06
External radius	0.5000000	0.4999922	7.80E-06

Table 5: Absolute error for the internal and external radius location between exact R_{ex} and numerical R_{num} solution.

3.5 The Saltzman Problem

A strong one–dimensional shock wave driven by a piston constitutes this test problem. As proposed in [53,56], the initial two–dimensional domain is a box $\Omega(0) = [0;1] \times [0;0.1]$, filled with an ideal gas. The piston is pushing the gas from the left to the right and it is moving with constant velocity $\mathbf{v}_p = (1,0)$. The computational domain is covered by a distorted unstructured mesh, composed of 100×10 elements along the boundaries, as reported in Figure 7.
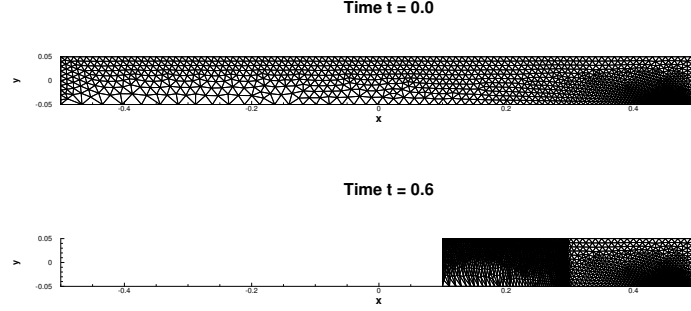
Figure 7: The initial and the final mesh for the Saltzman test problem.

The initial condition is given by an ideal gas at rest ($\mathbf{v}_0 = \mathbf{0}$) with a ratio of specific heats $\gamma = \frac{5}{3}$, an initial density of $\rho_0 = 0$ and an internal energy of $e_0 = 10^{-6}$, corresponding to an initial pressure of $p_0 = 6.67 \cdot 10^{-7}$.

As boundary conditions we set fixed slip wall boundaries on the upper and lower boundary, outflow on the right and a moving slip wall boundary condition for the piston on the left. According to [53] we use initially a Courant number of $CFL = 0.01$, in order to respect the geometric CFL condition and to avoid mesh elements crossing over. After this initial phase, the CFL condition is raised to its usual value of $CFL = 0.5$ in two space dimensions.

The exact solution is a one–dimensional infinite strength shock wave and it can be computed by solving the Riemann problem given in Table 6.

	Left state	Right state
ρ	1.0	1.0
u	1.0	-1.0
v	0.0	0.0
p	$6.67 \cdot 10^{-7}$	$6.67 \cdot 10^{-7}$

Table 6: One–dimensional Riemann problem for obtaining the exact solution of the Saltzman problem.

The details of the algorithm that computes the exact solution of the Riemann problem are given in the book of Toro [75]. The exact solution has then to be shifted by a certain quantity d to the right, corresponding to the movement of the piston during the time of the simulation t_f, i.e.

$$d = u_p \cdot t_f. \tag{3.19}$$

The exact post shock density is $\rho_e = 4.0$ and the final time is chosen to be $t_f = 0.6$, according to [53]. Therefore, the exact final shock location is at $x = 0.3$.

We use a Rusanov–type flux (2.44) and the solution is computed, both, for a third order and a fourth order scheme. We can notice a very good agreement regarding the

final density distribution, as depicted in Figure 8, and the solution for the velocity, where we observe a very sharp discontinuity at the shock location, as clearly shown in Figure 9. The decrease of the density near the piston in Figure 8 is due to the well known *wall–heating problem*, see [74]. One can notice from Figure 10 the progressive mesh compression at the inflow, where the piston is pushing, and the shock wave that is travelling through the domain, reaching its final location exactly according to the reference solution ($x=0.3$).
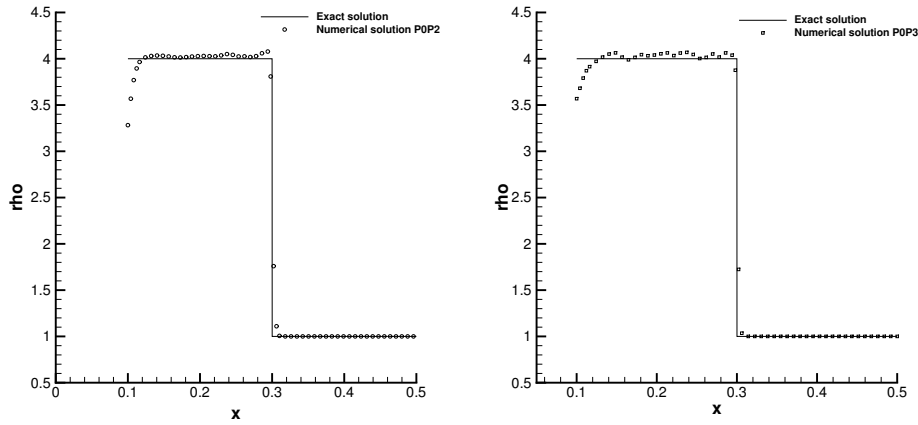
Figure 8: Numerical results for the density distribution of the Saltzman problem: third order scheme (left) and fourth order scheme (right). Both results are compared with the analytical solution, obtained by solving the Riemann problem described in Table 6.
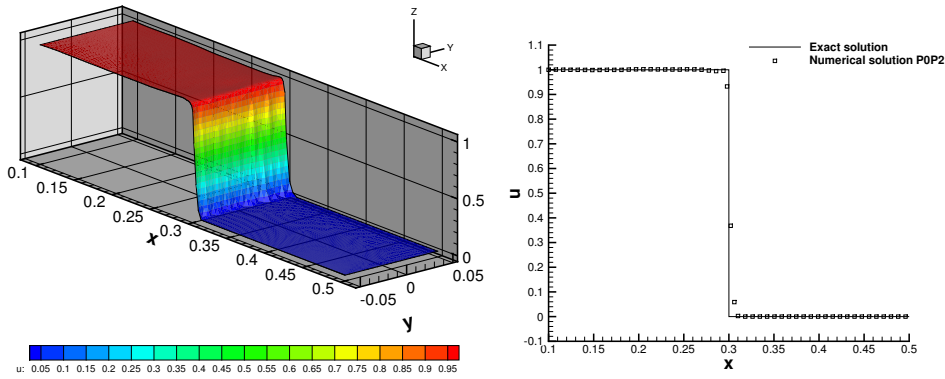
Figure 9: 3D visualization for the velocity distribution obtained by a third order finite volume scheme (left) and comparison with the analytical solution (right).
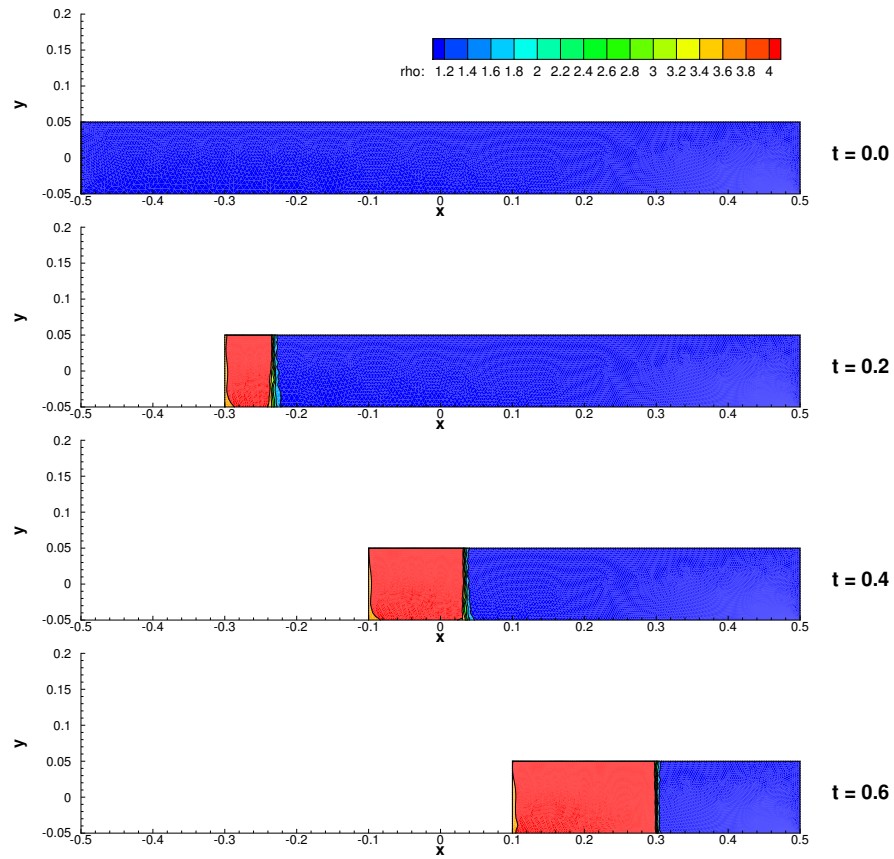
Figure 10: Progressive evolution of the numerical solution of the Saltzman problem (density distribution): the mesh is compressed by the piston on the left, while a shock wave is travelling towards the right. The final location coincides perfectly well with the exact solution.

3.6 Single Mach Reflection Problem

In this section we solve the single Mach reflection problem. It consists of a shock wave traveling at a shock Mach number of $M_s=1.7$ that hits a wedge of an angle $\alpha=25°$. Experimental reference data in form of Schlieren images as well as numerical reference solutions are shown, e.g. in [75]. The upstream density and pressure are $\rho_0=1$ and $p_0=1/\gamma$, respectively, where the ratio of specific heats is $\gamma=1.4$. In the following, the indices $_0$ and $_1$ will denote upstream and downstream states, respectively. Using the Rankine-Hugoniot conditions, we setup the computation with a shock wave initially centered at $x=-0.04$ that travels at $M_s=1.7$ to the right into a medium at rest. The wedge is defined by the triangle $\mathcal{W}_{25}$, composed of the vertices $(0,0)$, $(3.0,0)$ and $(3.0,1.398923)$. The initial computational domain $\Omega(0)=[-2;3]\times[0;2]\setminus\mathcal{W}_{25}$ is discretized with 177,440 triangles of characteristic size $h=1/100$ and a P_0P_2 WENO finite volume scheme is used. At the final time $t=1.2$ the exact shock location must be $x=2$. The upper and lower boundaries of the computational domain are discretized by slip walls. At the other boundaries Dirichlet boundary conditions consistent with the initial condition are imposed.
The results of the computation are depicted in Fig. 11, where the density contour lines are shown, as well as a zoom into the final mesh, which is very distorted along the slip line. This is to verify that our proposed high order unstructured finite volume algorithm can handle reasonably strong mesh deformations, which are a common feature of Lagrangian computations. For even stronger mesh deformations, as they occur in the double Mach reflection problem, a proper remeshing and projection strategy will be implemented in the future. The general flow features depicted in Fig. 11 agree very well with the computations and the experimental data shown in [75].

3.7 Two-dimensional Riemann Problems

In this section we solve a set of two–dimensional Riemann problems, whose initial conditions are given by

$$\mathbf{Q}(x,0)=\begin{cases} \mathbf{Q}_1 & \text{if} \quad x>0 \wedge y>0, \\ \mathbf{Q}_2 & \text{if} \quad x\leq 0 \wedge y>0, \\ \mathbf{Q}_3 & \text{if} \quad x\leq 0 \wedge y\leq 0, \\ \mathbf{Q}_4 & \text{if} \quad x>0 \wedge y\leq 0. \end{cases} \tag{3.20}$$

A large set of such two–dimensional Riemann problems has been presented in great detail in the paper by Kurganov and Tadmor [51]. The initial conditions for the three configurations presented in this article are listed in Table 7. The initial computational domain is defined as $\Omega(0)=[-0.5;0.5]\times[-0.5;0.5]$. The Lagrangian simulations are carried out with a third order one–step WENO scheme using an unstructured triangular mesh composed of 90,080 elements with an initial characteristic mesh spacing of $h=1/200$. The reference solution is computed with a high order Eulerian one–step WENO finite volume scheme as presented in [25, 27, 28], using a very fine mesh composed of 2,277,668 triangles with characteristic mesh spacing $h=1/1000$. In all cases, the Rusanov flux has
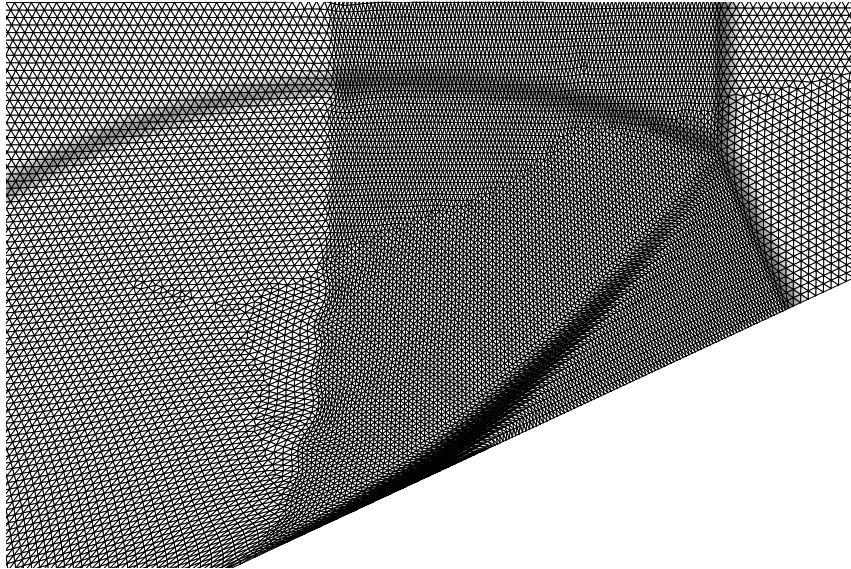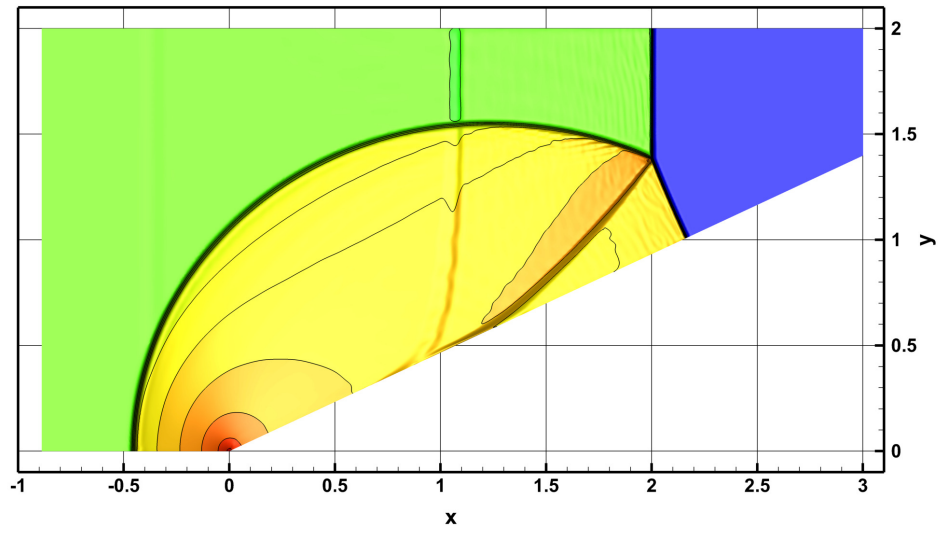
Figure 11: Top: 21 equidistant density contours from $\rho = 1.2$ to $\rho = 3$ for the single Mach reflection problem ($M_s = 1.7, \alpha = 25°$). Bottom: zoom into the mesh at the final time $t = 1.2$.

been used. The exact solutions of the one-dimensional Riemann problems are imposed as boundary conditions on the four boundaries of the domain. The results obtained with the Lagrangian-type scheme together with the final mesh and the Eulerian reference solution are depicted in Figures 12 - 13. We can note a very good qualitative agreement of the Lagrangian solution with the reference solution, as well as with the results published in [51].

Table 7: Initial conditions for the two–dimensional Riemann problems.

	Problem RP1 (Quadruple Sod problem)							
	$x \leq 0$				$x > 0$			
	ρ	u	v	p	ρ	u	v	p
$y > 0$	1.0	0.0	0.0	1.0	0.125	0.0	0.0	0.1
$y \leq 0$	0.125	0.0	0.0	0.1	1.0	0.0	0.0	1.0
	Problem RP2 (Configuration 2 in [51])							
	$x \leq 0$				$x > 0$			
	ρ	u	v	p	ρ	u	v	p
$y > 0$	0.5197	-0.7259	0.0	0.4	1.0	0.0	0.0	1.0
$y \leq 0$	1.0	-0.7259	-0.7259	1.0	0.5197	0.0	-0.7259	0.4
	Problem RP3 (Configuration 4 in [51])							
	$x \leq 0$				$x > 0$			
	ρ	u	v	p	ρ	u	v	p
$y > 0$	0.5065	0.8939	0.0	0.35	1.1	0.0	0.0	1.1
$y \leq 0$	1.1	0.8939	0.8939	1.1	0.5065	0.0	0.8939	0.35

4 Conclusions

We have developed a new high order two–dimensional Arbitrary–Lagrangian-Eulerian one–step WENO finite volume scheme on unstructured triangular meshes. The algorithm works for general hyperbolic balance laws with non stiff algebraic source terms. Several smooth and non–smooth test problems with shock waves, contact waves, shear waves and rarefactions have been simulated. The results have been compared with exact or numerical reference solutions in order to validate our approach. The algorithm was found to work properly in all cases and to be robust and accurate, too. The accuracy has been verified empirically via numerical convergence studies on a smooth test case with exact solution. Further work will concern the improvement of the presented algorithm in order to deal with stiff algebraic source terms, which is straightforward, see the work presented in [30]. Moreover, we plan to extend our scheme to three space dimensions in the more general framework of the new $P_N P_M$ method proposed in [25], where one can
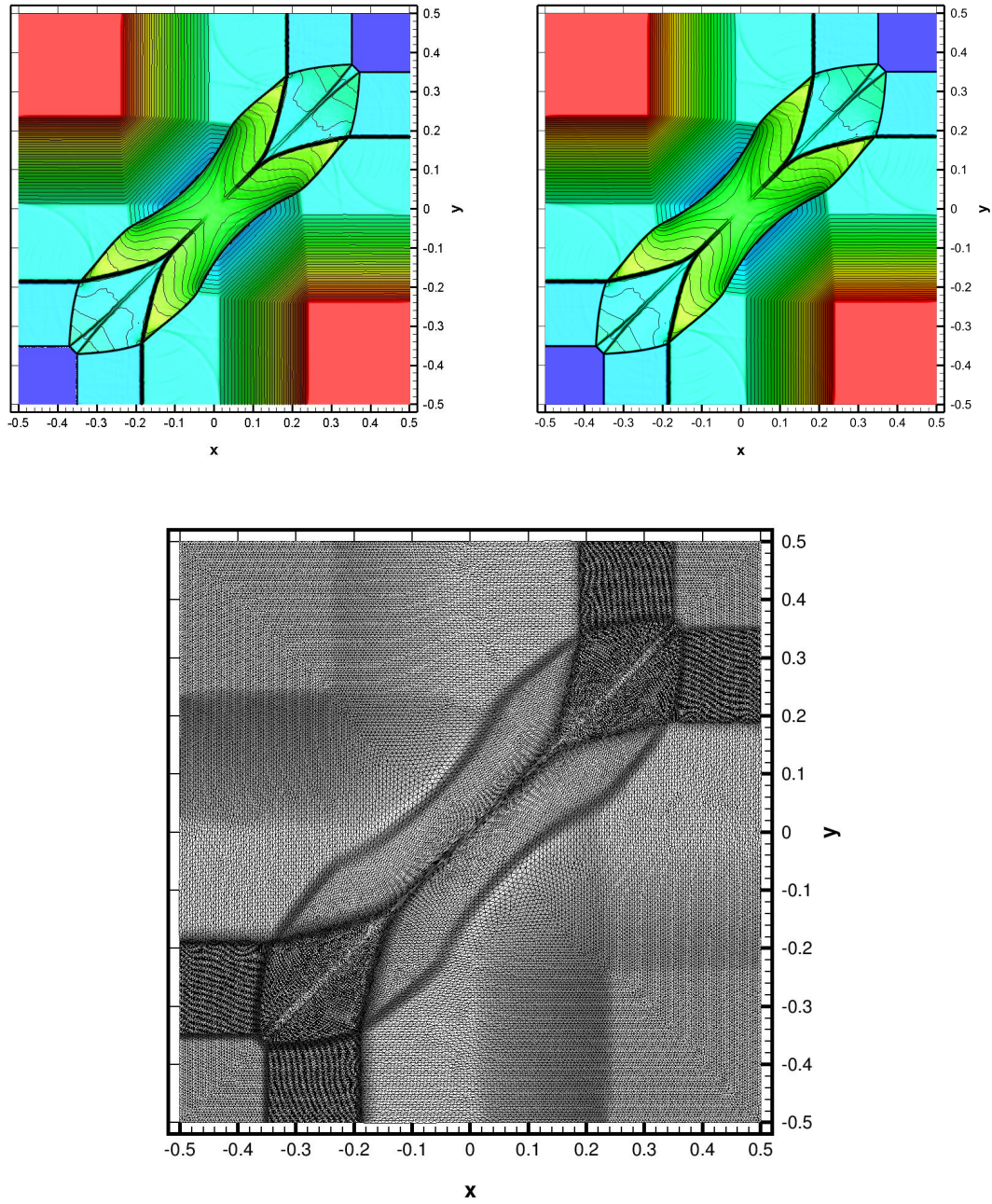
Figure 12: Top left: 50 equidistant density contour lines from $\rho = 0.14$ to $\rho = 0.98$ for the numerical solution obtained with our third order unstructured Lagrangian one–step WENO finite volume scheme for the two–dimensional Riemann problem RP1 at $t = 0.2$. Top right: Same contour lines for the reference solution. Bottom: computational mesh of the Lagrangian scheme at the final time $t = 0.2$.
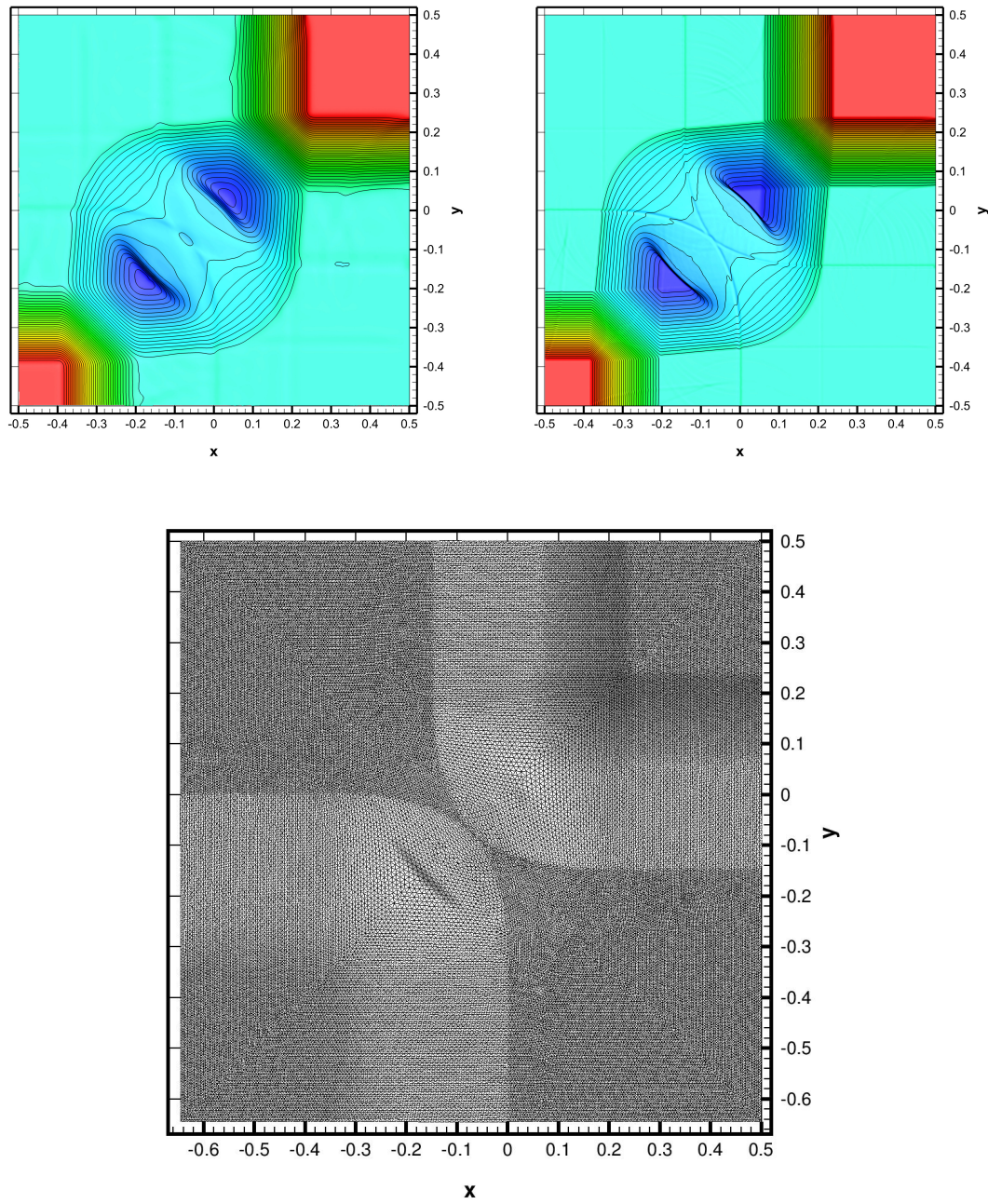
Figure 13: Top left: 51 equidistant density contour lines from $\rho = 0.27$ to $\rho = 0.98$ for the numerical solution obtained with our third order unstructured Lagrangian one–step WENO finite volume scheme for the two–dimensional Riemann problem RP2 at $t = 0.2$. Top right: Same contour lines for the reference solution. Bottom: computational mesh of the Lagrangian scheme at the final time $t = 0.2$.
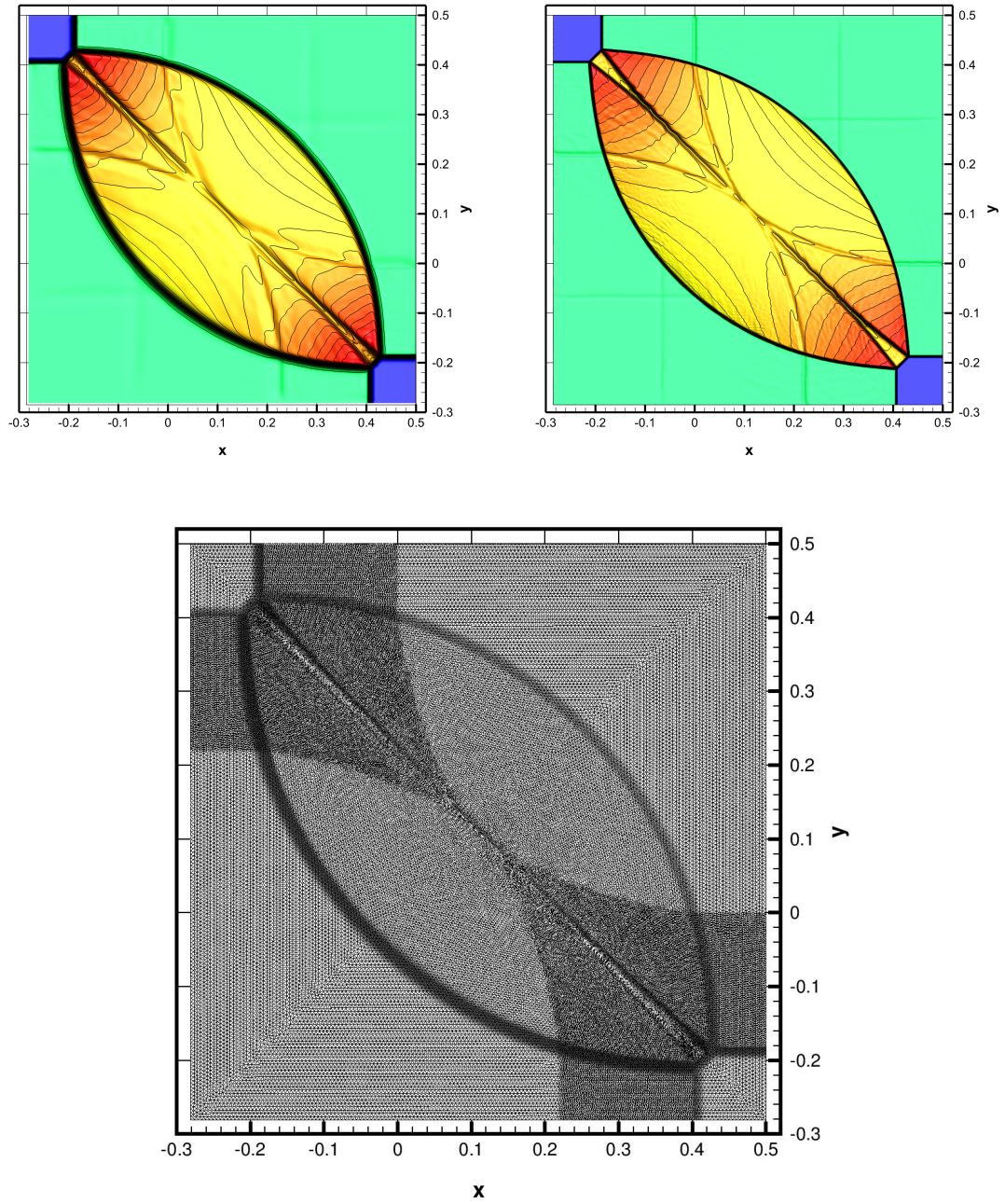
Figure 14: Top left: 51 equidistant density contour lines from $\rho = 0.53$ to $\rho = 1.9$ for the numerical solution obtained with a third order unstructured Lagrangian one–step WENO finite volume scheme for the two–dimensional Riemann problem RP3 at $t = 0.245$. Top right: Same contour lines for the reference solution. Bottom: computational mesh of the Lagrangian scheme at the final time $t = 0.245$.

deal with pure finite volume or pure finite element methods, or with a hybridization of both. Further work will also consist in a generalization to moving *curved* meshes as well as to hyperbolic PDE with *non–conservative* products.

Acknowledgments

References

[1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics*, 144:45–58, 1994.

[2] T. Aboiyar, E.H. Georgoulis, and A. Iske. Adaptive ADER Methods Using Kernel-Based Polyharmonic Spline WENO Reconstruction. *SIAM Journal on Scientific Computing*, 32:3251–3277, 2010.

[3] D.S. Balsara and C.W. Shu. Monotonicity perserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *J. Comput. Phys.*, 160:405–452, 2000.

[4] T.J. Barth and P.O. Frederickson. Higher order solution of the euler equations on unstructured grids using quadratic reconstruction. *28th Aerospace Sciences Meeting*, pages AIAA paper no. 90–0013, January 1990.

[5] M. Ben-Artzi and J. Falcovitz. A second-order godunov-type scheme for compressible fluid dynamics. *Journal of Computational Physics*, 55:1–32, 1984.

[6] D.J. Benson. Computational methods in lagrangian and eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99:235–394, 1992.

[7] W. Boscheri, M. Dumbser, and M. Righetti. A semi-implicit scheme for 3d free surface flows with high order velocity reconstruction on unstructured voronoi meshes. *International Journal for Numerical Methods in Fluids*. submitted to.

[8] A. Bourgeade, P. LeFloch, and P.A. Raviart. An asymptotic expansion for the solution of the generalized Riemann problem. Part II: application to the gas dynamics equations. *Annales de l'institut Henri Poincaré (C) Analyse non linéaire*, 6:437–480, 1989.

[9] E.J. Caramana, D.E. Burton, M.J. Shashkov, and P.P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *Journal of Computational Physics*, 146:227–262, 1998.

[10] G. Carré, S. Del Pino, B. Després, and E. Labourasse. A cell-centered lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics*, 228:5160 – 5183, 2009.

[11] V. Casulli. Semi-implicit finite difference methods for the two-dimensional shallow water equations. *Journal of Computational Physics*, 86:56–74, 1990.

[12] V. Casulli and R.T. Cheng. Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal of Numerical Methods in Fluids*, 15:629–648, 1992.

[13] J. Cesenek, M. Feistauer, J. Horacek, V. Kucera, and J. Prokopova. Simulation of compressible viscous flow in time-dependent domains. *Applied Mathematics and Computation*. DOI: 10.1016/j.amc.2011.08.077, in press.

[14] J. Cheng and C.W. Shu. A high order ENO conservative Lagrangian type scheme for the compressible Euler equations. *Journal of Computational Physics*, 227:1567–1596, 2007.

[15] J. Cheng and C.W. Shu. A cell-centered Lagrangian scheme with the preservation of symmetry and conservation properties for compressible fluid flows in two-dimensional cylindrical geometry. *Journal of Computational Physics*, 229:7191–7206, 2010.

[16] J. Cheng and C.W. Shu. Improvement on spherical symmetry in two-dimensional cylindrical coordinates for a class of control volume Lagrangian schemes. *Communications in Computational Physics*, 11:1144–1168, 2012.

[17] S. Clain, S. Diot, and R. Loubère. A high–order finite volume method for systems of conservation laws – Multi–dimensional Optimal Order Detection (MOOD). *Journal of Computational Physics*, 230:4028–4050, 2011.

[18] A. Claisse, B. Després, E.Labourasse, and F. Ledoux. A new exceptional points method with application to cell-centered lagrangian schemes and curved meshes. *Journal of Computational Physics*, 231:4324–4354, 2012.

[19] B. Cockburn, G. E. Karniadakis, and C.W. Shu. *Discontinuous Galerkin Methods*. Lecture Notes in Computational Science and Engineering. Springer, 2000.

[20] R. Courant, E. Isaacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Comm. Pure Appl. Math.*, 5:243–255, 1952.

[21] B. Després and C. Mazeran. Symmetrization of lagrangian gas dynamic in dimension two and multimdimensional solvers. *C.R. Mecanique*, 331:475–480, 2003.

[22] B. Després and C. Mazeran. Lagrangian gas dynamics in two-dimensions and lagrangian systems. *Archive for Rational Mechanics and Analysis*, 178:327–372, 2005.

[23] L. Dubcova, M. Feistauer, J. Horacek, and P. Svacek. Numerical simulation of interaction between turbulent flow and a vibrating airfoil. *Computing and Visualization in Science*, 12:207–225, 2009.

[24] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6:345–390, 1991.

[25] M. Dumbser, D.S. Balsara, E.F. Toro, and C.-D. Munz. A unified framework for the construction of one-step finite volume and discontinuous galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227:8209 – 8253, 2008.

[26] M. Dumbser, C. Enaux, and E.F. Toro. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *Journal of Computational Physics*, 227:3971 4001, 2008.

[27] M. Dumbser and M. Kaeser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221:693 – 723, 2007.

[28] M. Dumbser, M. Kaeser, V.A. Titarev, and E.F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226:204 – 243, 2007.

[29] M. Dumbser and E.F. Toro. On universal oshertype schemes for general nonlinear hyperbolic conservation laws. *Communications in Computational Physics*, 10:635 – 671, 2011.

[30] M. Dumbser, A. Uuriintsetseg, and O. Zanotti. On Arbitrary–Lagrangian–Eulerian One–Step WENO Schemes for Stiff Hyperbolic Balance Laws. *Communications in Computational Physics*, 14:301–327, 2013.

[31] M. Dumbser and O. Zanotti. Very high order PNPM schemes on unstructured meshes for the resistive relativistic MHD equations. *Journal of Computational Physics*, 228:6991–7006, 2009.

[32] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*,

152:457–492, 1999.

[33] R.P. Fedkiw, T. Aslam, and S. Xu. The Ghost Fluid method for deflagration and detonation discontinuities. *Journal of Computational Physics*, 154:393–427, 1999.

[34] M. Feistauer, J. Horacek, M. Ruzicka, and P. Svacek. Numerical analysis of flow-induced nonlinear vibrations of an airfoil with three degrees of freedom. *Computers and Fluids*, 49:110–127, 2011.

[35] M. Feistauer, V. Kucera, J. Prokopova, and J. Horacek. The ALE discontinuous Galerkin method for the simulatio of air flow through pulsating human vocal folds. *AIP Conference Proceedings*, 1281:83–86, 2010.

[36] A. Ferrari. SPH simulation of free surface flow over a sharp-crested weir. *Advances in Water Resources*, 33:270–276, 2010.

[37] A. Ferrari, M. Dumbser, E.F. Toro, and A. Armanini. A New Stable Version of the SPH Method in Lagrangian Coordinates. *Communications in Computational Physics*, 4:378–404, 2008.

[38] A. Ferrari, M. Dumbser, E.F. Toro, and A. Armanini. A new 3D parallel SPH scheme for free surface flows. *Computers & Fluids*, 38:1203–1217, 2009.

[39] A. Ferrari, L. Fraccarollo, M. Dumbser, E.F. Toro, and A. Armanini. Three–dimensional flow evolution after a dambreak. *Journal of Fluid Mechanics*, 663:456–477, 2010.

[40] A. Ferrari, C.D. Munz, and B. Weigand. A high order sharp interface method with local timestepping for compressible multiphase flows. *Communications in Computational Physics*, 9:205–230, 2011.

[41] P. Le Floch and P.A. Raviart. An asymptotic expansion for the solution of the generalized Riemann problem. Part I: General theory. *Annales de l'institut Henri Poincaré (C) Analyse non linéaire*, 5:179–207, 1988.

[42] R.W. Healy and T.F. Russel. Solution of the advection-dispersion equation in two dimensions by a finite-volume eulerian-lagrangian localized adjoint method. *Advances in Water Resources*, 21:11–26, 1998.

[43] A. Hidalgo and M. Dumbser. ADER schemes for nonlinear systems of stiff advectiondiffusionreaction equations. *Journal of Scientific Computing*, 48:173–189, 2011.

[44] C. Hirt, A. Amsden, and J. Cook. An arbitrary lagrangianeulerian computing method for all flow speeds. *Journal of Computational Physics*, 14:227253, 1974.

[45] C. Hu and C.W. Shu. A high-order weno finite difference scheme for the equations of ideal magnetohydrodynamics. *Journal of Computational Physics*, 150:561 – 594, 1999.

[46] C.S. Huang, T. Arbogast, and J. Qiu. An eulerian-lagrangian weno finite volume scheme for advection problems. *Journal of Computational Physics*, 231:4028–4052, 2012.

[47] G.S. Jiang and C.W. Shu. Efficient implementation of weighted eno schemes. *Journal of Computational Physics*, pages 202 – 228, 1996.

[48] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp Element Methods in CFD*. Oxford University Press, 1999.

[49] M. Käser and A. Iske. Ader schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205:486 – 508, 2005.

[50] R.E. Kidder. Laser-driven compression of hollow shells: power requirements and stability limitations. *Nucl. Fus.*, 1:3 – 14, 1976.

[51] A. Kurganov and E. Tadmor. Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers. *Numerical Methods for Partial Differential Equations*, 18:584–608, 2002.

[52] M. Lentine, Jón Tómas Grétarsson, and R. Fedkiw. An unconditionally stable fully conser-

vative semi-lagrangian method. *Journal of Computational Physics*, 230:2857–2879, 2011.

[53] W. Liu, J. Cheng, and C.W. Shu. High order conservative Lagrangian schemes with LaxWendroff type time discretization for the compressible Euler equations. *Journal of Computational Physics*, 228:8872–8891, 2009.

[54] P.-H. Maire. A high-order one-step sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids. *Computers and Fluids*, 46(1):341–347, 2011.

[55] P.-H. Maire. A unified sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids. *International Journal for Numerical Methods in Fluids*, 65:12811294, 2011.

[56] P.H. Maire. A high-order cell-centered lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *Journal of Computational Physics*, 228:2391 – 2425, 2009.

[57] P.H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered lagrangian scheme for two-dimensional compressible flow problems. *SIAM Journal on Scientific Computing*, 29:17811824, 2007.

[58] J.J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110:399–406, 1994.

[59] W. Mulder, S. Osher, and J.A. Sethian. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, 100:209–228, 1992.

[60] C.D. Munz. On Godunov–type schemes for Lagrangian gas dynamics. *SIAM Journal on Numerical Analysis*, 31:17–42, 1994.

[61] C. Olliver-Gooch and M. Van Altena. A high-order-accurate unstructured mesh finite-volume scheme for the advectiondiffusion equation. *Journal of Computational Physics*, 181:729 – 752, 2002.

[62] A. Lpez Ortega and G. Scovazzi. A geometrically–conservative, synchronized, flux–corrected remap for arbitrary Lagrangian–Eulerian computations with nodal finite elements. *Journal of Computational Physics*, 230:6709–6741, 2011.

[63] S. Osher and J.A. Sethian. Fronts propagating with curvature–dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[64] J.S. Peery and D.E. Carroll. Multi-material ale methods in unstructured grids,. *Computer Methods in Applied Mechanics and Engineering*, 187:591–619, 2000.

[65] Jing-Mei Qiu and Chi-Wang Shu. Conservative high order semi-lagrangian finite difference weno methods for advection in incompressible flow. *Journal of Computational Physics*, 230:863–889, 2011.

[66] K. Riemslagh, J. Vierendeels, and E. Dick. An arbitrary lagrangian-eulerian finite-volume method for the simulation of rotary displaecment pump flow. *Applied Numerical Mathematics*, 32:419–433, 2000.

[67] G. Scovazzi. Lagrangian shock hydrodynamics on tetrahedral meshes: A stable and accurate variational multiscale approach. *Journal of Computational Physics*, 231:8029–8069, 2012.

[68] R.W. Smith. AUSM(ALE): a geometrically conservative arbitrary lagrangianeulerian flux splitting scheme. *Journal of Computational Physics*, 150:268286, 1999.

[69] A.H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.

[70] V.A. Titarev and E.F. Toro. ADER: Arbitrary high order Godunov approach. *Journal of Scientific Computing*, 17:609–618, 2002.

[71] V.A. Titarev and E.F. Toro. ADER schemes for three-dimensional nonlinear hyperbolic systems. *Isaac Newton Institute for Mathematical Sciences Preprint Series*, 2003.

[72] V.A. Titarev, P. Tsoutsanis, and D. Drikakis. WENO schemes for mixed–element unstruc-

tured meshes. *Communications in Computational Physics*, 8:585–609, 2010.

[73] E. F. Toro and V. A. Titarev. Derivative Riemann solvers for systems of conservation laws and ADER methods. *Journal of Computational Physics*, 212(1):150–165, 2006.

[74] E.F. Toro. Anomalies of conservative methods: analysis, numerical evidence and possible cures. *International Journal of Computational Fluid Dynamics*, 11:128–143, 2002.

[75] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction.* Springer, 2009.

[76] P. Tsoutsanis, V.A. Titarev, and D. Drikakis. WENO schemes on arbitrary mixed-element unstructured meshes in three space dimensions. *Journal of Computational Physics*, 230:1585–1601, 2011.

[77] J. J. W. van der Vegt and H. van der Ven. Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows I. general formulation. *Journal of Computational Physics*, 182:546–585, 2002.

[78] H. van der Ven and J. J. W. van der Vegt. Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows II. efficient flux quadrature. *Comput. Methods Appl. Mech. Engrg.*, 191:4747–4780, 2002.

[79] J. von Neumann and R.D. Richtmyer. A method for the calculation of hydrodynamics shocks. *Journal of Applied Physics*, 21:232–237, 1950.

[80] Y.T. Zhang and C.W. Shu. Third order WENO scheme on three dimensional tetrahedral meshes. *Communications in Computational Physics*, 5:836–848, 2009.